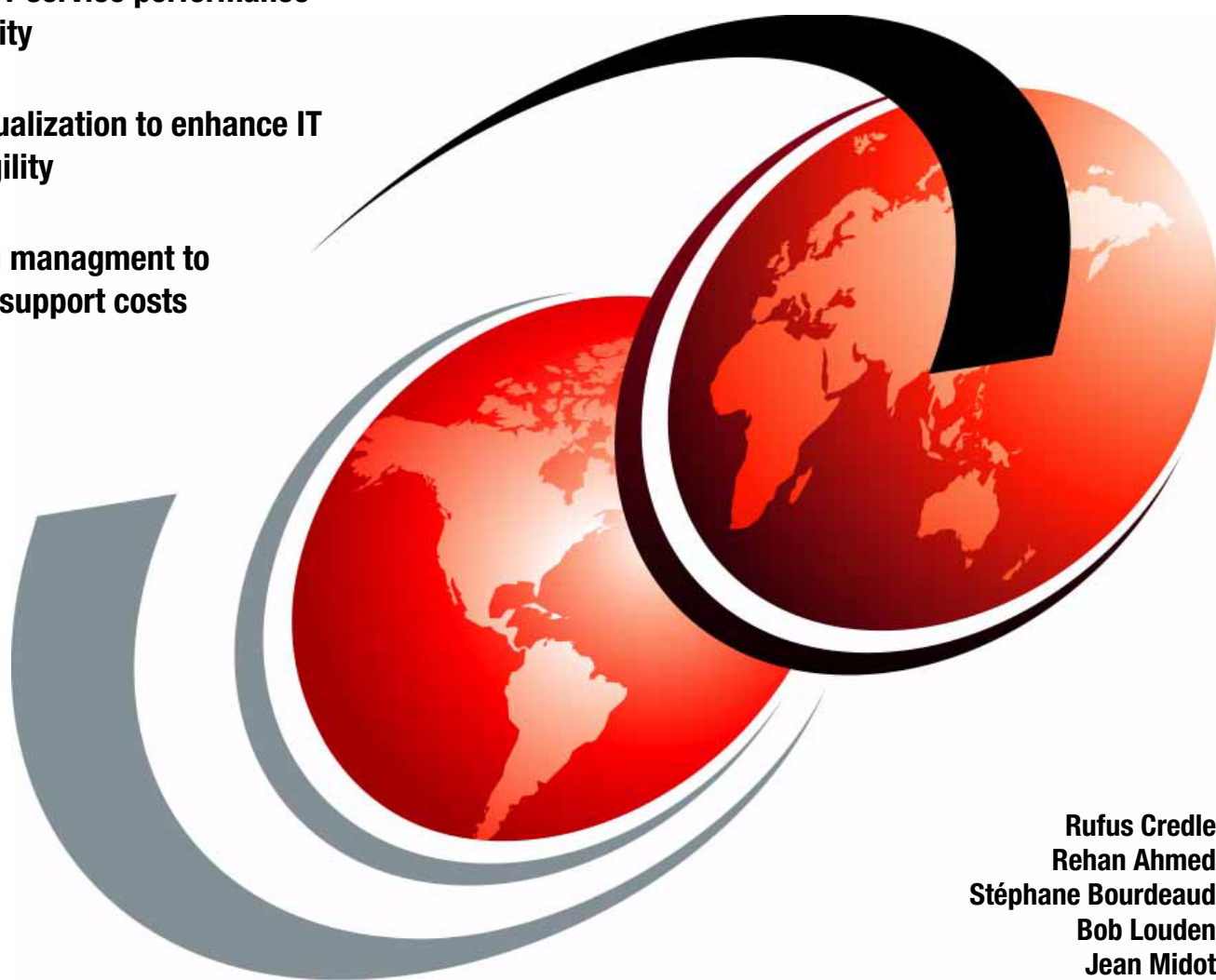


Implementing a VM-Aware Network Using VMready

Ensuring IT service performance and security

Using virtualization to enhance IT service agility

Improving management to reduce IT support costs



Rufus Credle
Rehan Ahmed
Stéphane Bourdeaud
Bob Louden
Jean Midot

Redbooks



International Technical Support Organization

Implementing a VM-Aware Network Using VMready

August 2012

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (August 2012)

This edition applies to IBM Networking OS 6.9 and the following IBM System Networking hardware:
IBM BNT Rack Switch G8264 (7309HC3/4)
BNT Virtual Fabric 10 Gb Switch Module for IBM BladeCenter (46C7191)

© Copyright International Business Machines Corporation 2012. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
The team who wrote this book	ix
Now you can become a published author, too!	xi
Comments welcome	xi
Stay connected to IBM Redbooks	xi
Part 1. Virtualization and VM-aware networking overview	1
Chapter 1. What you need to know about system virtualization	3
1.1 Virtualization overview	4
1.2 Server virtualization	4
1.3 Characteristics of common virtualization environments	8
Chapter 2. Introducing VMready	11
2.1 Network support of system virtualization	12
2.2 VMready overview	13
2.3 VMready with Edge Virtual Bridging (IEEE 802.1Qbg)	14
2.3.1 VDP: Automating network switch configuration	14
2.3.2 VEPA: An alternative to vSwitches	16
2.3.3 Edge Virtual Bridging (IEEE 802.1Qbg) for VMware	17
2.4 VMready without Edge Virtual Bridging	18
2.4.1 Virtual machine identities	19
2.4.2 Grouping	20
2.4.3 NMotion	20
Chapter 3. Management for VMready	23
3.1 IBM System Networking Element Manager Overview	24
3.1.1 IBM System Networking Element Manager component	24
3.1.2 IBM Tivoli OMNIBus and Network Manager	24
3.1.3 IBM Tivoli Netcool Configuration Manager	25
3.2 Configuring VMready with System Networking Element Manager	25
3.2.1 Configuring VMready for Edge Virtual Bridging (IEEE 802.1Qbg)	26
3.2.2 Configuring VMready when not using Edge Virtual Bridging	33
3.3 Monitoring and troubleshooting your VMready installation	40
3.3.1 Monitoring VMready with Edge Virtual Bridging (IEEE 802.1Qbg)	41
3.3.2 Monitoring VMready when not using Edge Virtual Bridging	41
Part 2. Implementing a VM-aware network	43
Chapter 4. Implementing VMready to support VMware	45
4.1 Overview of VMready for VMware	46
4.1.1 VMware networking	46
4.1.2 VMready capabilities in a VMware environment	47
4.1.3 VMready implementation overview	47
4.2 Implementation scenario	49
4.2.1 Example IT policies	49
4.2.2 Environment overview	49

4.2.3	Initial configuration	50
4.2.4	Target scenario	54
4.3	Implementing VMready	55
4.3.1	Definitions	55
4.3.2	Configuration workflow	56
4.3.3	Entering configuration mode	56
4.3.4	Enabling VMready	57
4.3.5	Defining vmprofiles	59
4.3.6	Defining group ACLs (vmaps)	59
4.3.7	Applying vmprofiles to vmgroups	59
4.3.8	Applying vmaps to vmgroups	60
4.3.9	Adding VMs to vmgroups	60
4.3.10	Exporting vmprofiles to ESX hosts	60
4.3.11	Assigning per-VM bandwidth policy	62
4.3.12	Post implementation review	62
4.3.13	NMotion	66
4.3.14	Testing Policy Compliance	68
4.3.15	VMready summary	71
4.4	Implementing VMready with IEEE 802.1Qbg support	72
4.4.1	Components required to enable EVB for VMware	72
4.4.2	Target scenario	73
4.4.3	EVB installation workflow	74
4.4.4	EVB installation steps	75
4.4.5	EVB configuration workflow	83
4.4.6	EVB configuration steps	85
4.4.7	Post implementation review	93
4.4.8	NMotion	99
4.4.9	Testing policy compliance	101
Chapter 5. Implementing VMready to support PowerVM.		105
5.1	Overview of VMready for PowerVM	106
5.1.1	A little about PowerVM networking	106
5.1.2	The example system	106
5.1.3	The example scenario	106
5.2	Initial configuration	107
5.2.1	Hardware	107
5.2.2	VIO Server	107
5.2.3	Shared storage	108
5.2.4	Network	109
5.2.5	LPARs	110
5.2.6	Switch configuration	110
5.3	Initial tests	111
5.3.1	Web connection test	112
5.3.2	Bandwidth test	114
5.4	Enabling VMready and implementing network policies	114
5.4.1	Enabling VMready	114
5.4.2	Enabling locally administered MAC addresses	116
5.4.3	Creating VM Groups	116
5.4.4	Applying access control lists	123
5.4.5	Applying traffic shaping	126
5.5	Validating the network policies	128
5.5.1	Validating the ACL policy	128
5.5.2	Validating the traffic shaping policy	130

5.6 Validating Nmotion	130
5.6.1 Validating the ACL policy	133
5.6.2 Validating the traffic shaping policy.	134
Chapter 6. Implementing VMready to support KVM.	135
6.1 Overview of VMready for KVM	136
6.1.1 A little about KVM networking	136
6.1.2 VMready capabilities in a KVM environment.	137
6.1.3 VMready terminology	137
6.1.4 Example setup	137
6.1.5 The example scenario.	138
6.2 Implementation	139
6.2.1 Step 1: What the environment looks like without VMready.	140
6.2.2 Step 2: Configuring VMready	141
6.2.3 Step 3: Creating a VM Group and assigning virtual machines to it.	145
6.2.4 Step 4: Creating a VMAP to enforce an Access Control List	151
6.2.5 Step 5: Adding a virtual machine bandwidth policy.	157
6.2.6 Step 6: Demonstrating NMotion	160
6.3 Implementing IEEE 802.1Qbg with VMready	165
6.3.1 Step 1: Configuring the physical switch	167
6.3.2 Step 2: Configuring the IBM System Networking Element Manager.	169
6.3.3 Step 3: Configuring the hypervisor	173
6.3.4 Verifying the EVB configuration	178
6.3.5 Troubleshooting EVB	179
6.3.6 Implementing network policies with EVB	189
6.3.7 Edge Virtual Bridging for KVM Frequently Asked Questions	194
6.4 Conclusion	198
Chapter 7. Implementing VMready to support Hyper-V and other virtualization environments	199
7.1 Overview of VMready for Hyper-V	200
7.1.1 A little about Hyper-V networking	200
7.1.2 Example setup	200
7.1.3 The example scenario.	200
7.2 Initial configuration	201
7.2.1 Hardware	201
7.2.2 Hypervisor.	202
7.2.3 Shared storage	202
7.2.4 Network.	203
7.2.5 Virtual machines	204
7.2.6 Switch configuration	207
7.3 Initial tests	208
7.3.1 Web connection tests	208
7.3.2 Bandwidth tests.	210
7.4 Enabling VMready and implementing network policies	210
7.4.1 Enabling VMready.	211
7.4.2 Creating the VM Groups	214
7.4.3 Applying access control lists.	222
7.4.4 Applying traffic shaping.	225
7.5 Validating the network policies	227
7.5.1 Validating the ACL policy	227
7.5.2 Validating the traffic shaping policy.	229
7.6 Validating Nmotion	230
7.6.1 Validating the ACL policy	232

7.6.2 Validating the traffic shaping policy	233
Related publications	235
IBM Redbooks	235
Online resources	235
Help from IBM	237

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features addressed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	Power Systems™	System x®
BladeCenter®	POWER6®	System z®
BNT®	POWER7®	Tivoli®
Global Technology Services®	PowerVM®	VMready®
IBM®	POWER®	WebSphere®
Netcool®	Redbooks®	z/VM®
NMotion®	Redpapers™	
POWER Hypervisor™	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

AMD-V, the AMD Arrow logo, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

QLogic, and the QLogic logo are registered trademarks of QLogic Corporation. SANblade is a registered trademark in the United States.

Solaris, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Virtualization allows administrators to create virtual versions of resources such as a hardware platform, operating system, storage device, or network resource. This process allows the creation of multiple versions of a resource on a single physical machine. This configuration allows the advantages of multiple resources while simplifying management tasks and improving use of physical resources. In this way, virtualization can be used to enhance IT service performance, scalability, efficiency, availability, and security. Configuring these virtual machines requires the use of hypervisor software.

However, network switches are not aware of virtual machines. If you run each virtual machine on a dedicated set of servers, performance and security needs can be met by configuring the network settings for each servers. However, this nullifies the main benefits of virtualization. You can get better IT service performance, scalability, efficiency, and availability by creating multiple virtual resources on the same server. For this configuration, you need a way to apply unique network settings for each virtual resource.

IBM® VMready® is a software solution that supports open standards virtualization based on IEEE 802.1Qbg Edge Virtual Bridging. It allows administrators to create groups of virtual machines, administer them from a central location, and migrate them. VMready works with all major hypervisor software, including VMware, Microsoft Hyper-V, Linux Kernel-based Virtual Machine (KVM), Citrix XenServer, or IBM PowerVM®. It requires no proprietary tagging or changes to the hypervisor software. This IBM Redbooks® publication helps IT systems and networking professionals to understand IBM VMready technology options. It includes instructions on how to install, tailor, and configure VMready networking solutions.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Rufus Credle is a Certified Consulting IT Specialist at the ITSO, Raleigh Center. In his role as project leader, he conducts residencies and develops IBM Redbooks and Redpapers™. Subjects include network operating systems, enterprise resource planning (ERP) solutions, voice technology, high availability, clustering solutions, web application servers, pervasive computing, IBM and OEM e-business applications, WebSphere® Commerce, IBM industry technology, System x®, and IBM BladeCenter®. Rufus' various positions during his IBM career include assignments in administration and asset management, systems engineering, sales and marketing, and IT services. He has a BS degree in Business Management from Saint Augustine's College. Rufus has been employed at IBM for 31 years.

Rehan Ahmed is a Client Technical Specialist based in Sydney, Australia. He has 6 years of experience in managing System x and BladeCenter infrastructures. He holds a Bachelor's degree in Computer Science from the University of New South Wales. He has held positions of subject matter expert and account technical escalation point for over 10 IBM Global Services engagements. Over his career he has supported server infrastructures for more than 25 IBM clients, implemented virtualization proof of concepts, and delivered technology briefings to both technical and business audiences.

Stéphane Bourdeaud is a certified infrastructure architect for the IBM Global Technology Services® Services Delivery organization. He is based in Aubière, France. Stéphane has

over 15 years of experience in the IT industry and focuses primarily on virtualization technologies on Intel platforms. He also works for the Global Technology Services Delivery Technology and Engineering organization, where he is a member of the core team for the Virtualization and Distributed Server Management Specialty Area. In this capacity, he develops virtualization reference architectures and strategic outsourcing global offerings. He is a VMware Certified Professional and teaches a class on virtualization at the Institut Supérieur d'Informatique de Modélisation et de leurs Applications (ISIMA), an engineering school in Clermont Ferrand, France.

Bob Louden is a Consulting IT Specialist on the IBM Techline team responsible for technical sales support for IBM System Networking offerings. Bob holds a BS in Computer Science from Virginia Tech, and an MS in Computer and Communications Science from the University of Michigan. Bob has twenty-nine years of experience with IBM in roles including product development, sales, technical sales support, and consulting. He helps clients apply technology solutions to their business problems.

Jean Midot is a System x IT Specialist based in Montpellier, France, who has significant experience in installing, configuring and maintaining VMware environments. Jean also brings deep knowledge of Citrix Xen, Hyper-V, System x, BladeCenter, and networking. He is specialized in implementing and maintaining, through automated processes, cloud computing environments composed of virtual systems and physical hardware.

Thanks to the following people for their contributions to this project:

Special thanks to Pushkar Patil, who coordinated our lab environment and provided a technical support focal point for our implementations.

And to Rakesh Saha, and Jay Kidambi, for their technical leadership.

Larry Bailey
Stephan Benny
Amitabha Biswas
Deanna Brown
David Iles
Scott Irwin
Jeff Jaurigui
David Kasberg
Vivek Kashyap
Karen Lawrence
Scott Lorditch
Bob Nevins
Thomas Parker
Renato Recio
Rakesh Sharma
Gerhard Stenzel
Kishore Karolill
Thomas-Mich Richter
Tim Sutherland
David Watts
IBM

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Part 1

Virtualization and VM-aware networking overview

This part introduces virtualization, VM-aware networking, and management of system virtualization environments. The focus is on the technologies and tools at the intersection between virtualized systems and the networks that must support them. This part includes the following chapters:

- ▶ What you need to know about system virtualization
- ▶ Introducing VMready
- ▶ Management for VMready



What you need to know about system virtualization

A recent search on Amazon.com yielded almost 750 books on the various system virtualization products and technologies. Rather than trying to address virtualization technologies comprehensively, this chapter addresses *what you need to know* about system virtualization to implement a VM-aware network. This chapter includes the following sections:

- ▶ Virtualization overview
- ▶ Server virtualization
- ▶ Characteristics of common virtualization environments

1.1 Virtualization overview

In the context of information technology, virtualization is the approach of making one resource look like many or many resources look like one. For example, *virtual machines* have long been used to make a single computer look like many. Similarly, *virtual storage* is used to divide up the real storage in a computer among the virtual machines. Conversely, Grid Computing makes many independent computers work together as a single virtual computer. Networked storage (storage area network (SAN) and network-attached storage (NAS) technologies) likewise make many separate units of storage display as one large pool. And, in networking, *Virtual LANs* take a single Local Area Network (LAN) and create logically isolated sub-LANs (usually for performance or security purposes).

Virtualization allows you to enhance IT service performance and scalability, achieving greater capacity (server MIPS, storage GB, network K/M/Gbps) than is available from a single instance. Efficiency is improved through consolidation of workloads on under-utilized resources (storage, servers, or networks). This consolidation improves resource utilization and reduces overall hardware, software, and management costs. Virtualization can also be used to enhance availability and security. You can do so by placing untrusted applications such as Java applets into a controlled environment, and by isolating networks and servers into “zones” of trust.

Dividing a single computer system into multiple virtual machines can be accomplished through clever programming. Combining multiple resources into a single “virtual” system, however, requires cross-system cooperation (standards) and communications (networks). This is especially true when the multiple systems are heterogeneous. The emergence of the TCP/IP networking standards and the additional standards built upon them, including XML, SOAP, and WSDL, dramatically increase available network capacity. These standards enable new levels of virtualization.

Many more resources are being virtualized than just the storage and processing power of a single computer. Essentially, you can build high-capacity, highly available, and efficient virtualized computing systems, storage pools, and networks by combining the capacities of the underlying resources. These resources can be divided across business, security, or other logical domains into virtualized subsystems. You can view IT resources as a whole and dynamically (“on demand”). You can use this view to shift those resources to where they are most needed. Ultimately, you can achieve a much tighter coupling of business policy and IT resource management.

1.2 Server virtualization

With *server virtualization*, a hypervisor acts as an intermediary between the operating system and the hardware to achieve virtual machine separation. These operating systems are called *guests* or *virtual machines* (VMs). The hypervisor provides hardware emulation and manages the allocation of system resources among the VMs.

The hypervisor layer is installed on top of the hardware. This hypervisor manages access to hardware resources. The guest systems are then installed on top of the hypervisor as shown in Figure 1-1. This approach is also called *bare-metal* virtualization. It allows the guest operating system to be unaware that it is operating in a virtual environment. This approach does not require any modification of the guest operating system.

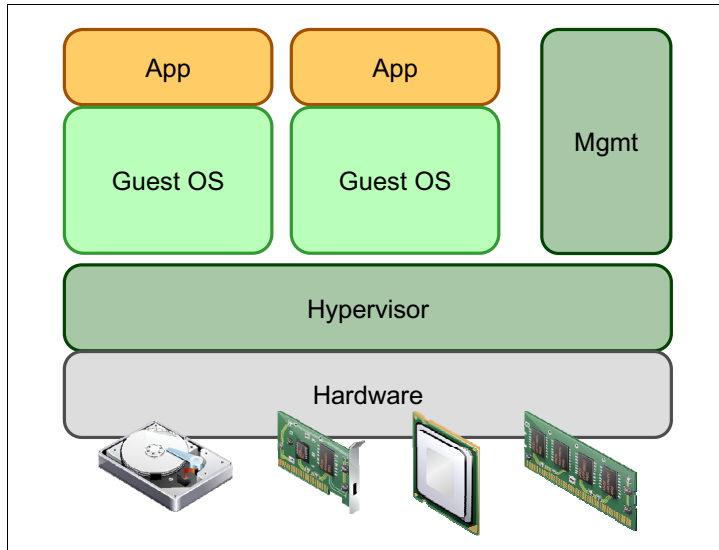


Figure 1-1 Server virtualization architecture

Server virtualization consolidates workload into fewer, more-powerful, servers. Likewise, however, the network bandwidth required to support those servers is consolidated as well. This configuration results in fundamental change to the underlying assumptions of the existing network design. This change can cause congestion and poor application performance, as shown in Figure 1-2.

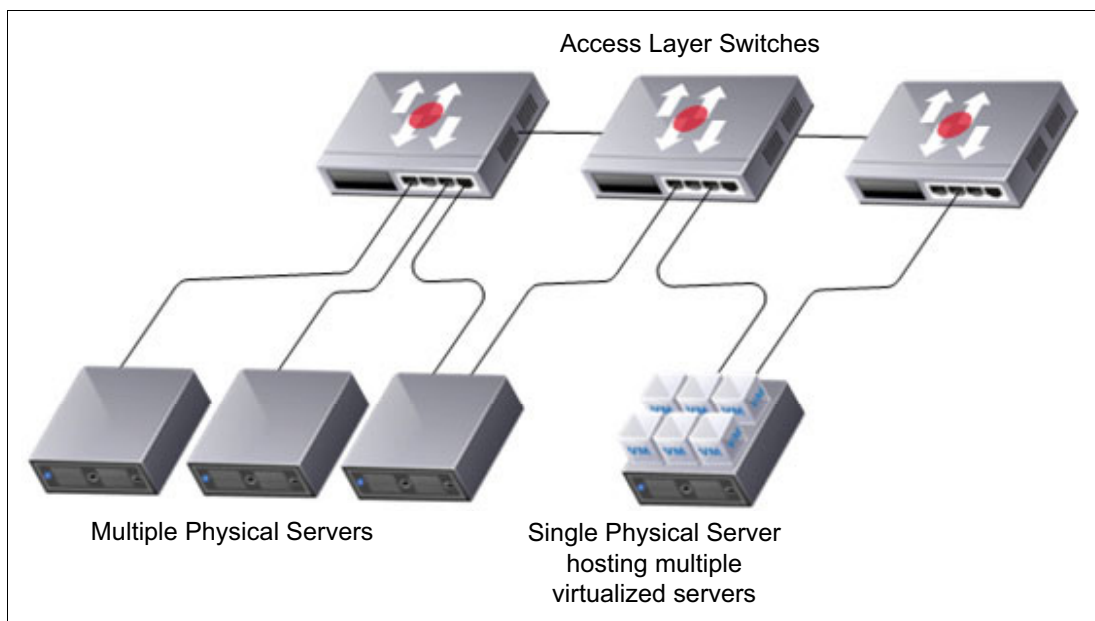


Figure 1-2 Network access requirements

Network interface sharing

Logical NIC sharing, as shown in Figure 1-3, allows each operating system to send packets to a single physical NIC. Each operating system has its own IP address. The server manager software typically has an additional IP address for configuration and management. A requirement of this solution is that all guest OSs must be in the same Layer 2 domain (subnet). Each guest OS must be assigned an IP address and a Media Access Control (MAC) address. Because the number of guest OSs that can exist on one platform is relatively small, the MAC address can be a modified version of the NIC burned-in MAC address. The IP addresses can consist of a small block of addresses in the same IP subnet. One additional IP address is used for the management console of the platform.

Features to manage quality of service (QoS) and load balancing to the physical NIC from the guest OSs are limited. In addition, any traffic from Guest OS1 destined to Guest OS2 travels out to a connected switch. It then returns along the same physical connection, which can add an extra load on the Ethernet connection.

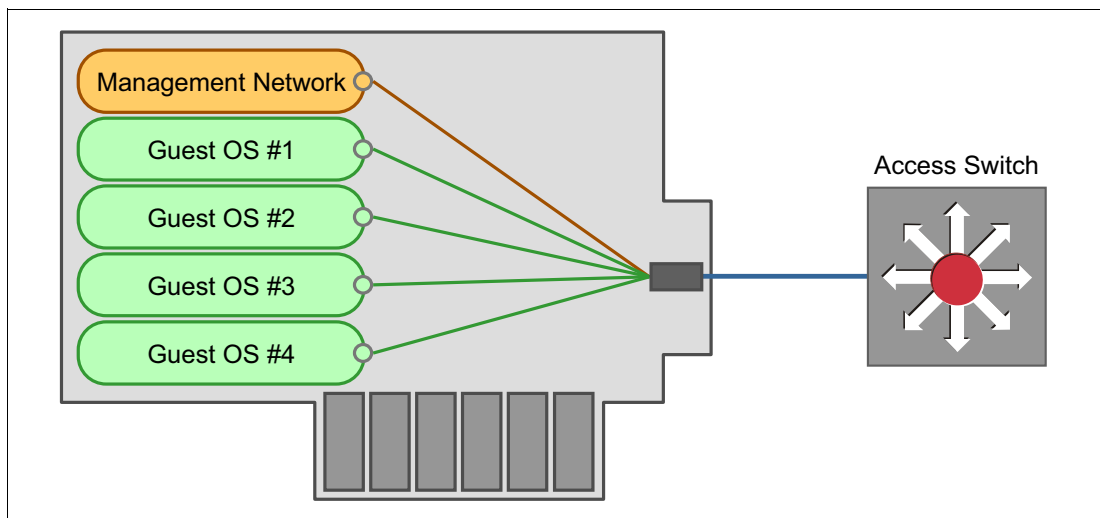


Figure 1-3 NIC sharing

Virtual NIC technology: Virtual switching

Virtual NIC (vNIC) technology enables each server to have a virtual NIC that connects to a virtual switch (vSwitch). This approach allows each operating system to exist in a separate Layer 2 domain. The connection between the virtual switch and the physical NIC then becomes an 802.1Q trunk. The physical connection between the physical NIC and the physical switch is also an 802.1Q trunk, as shown in Figure 1-4.

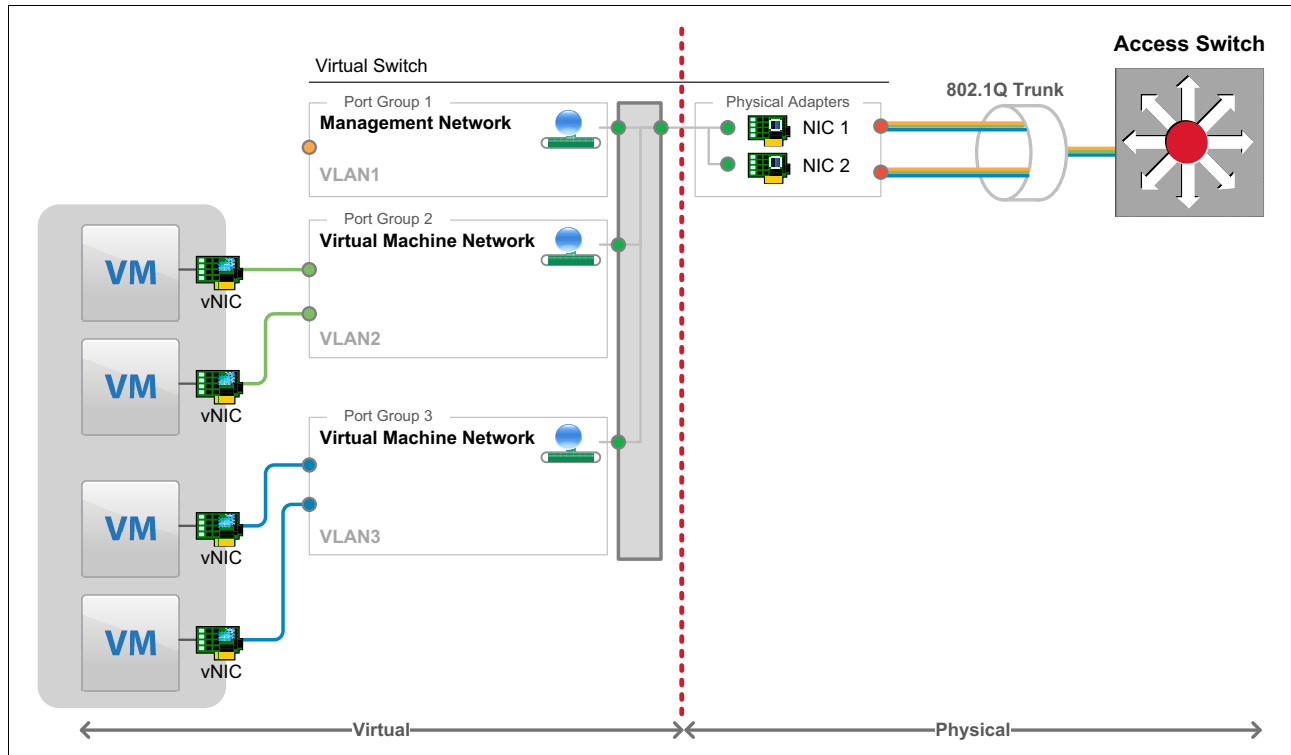


Figure 1-4 Virtual switching

A Layer 3 implementation of this feature allows traffic destined for a server on the same platform to be routed between virtual LANs (VLANs) within the host platform. This configuration prevents the traffic from going out of and then back into the Ethernet connection.

A Layer 2 implementation of this feature effectively makes the physical Layer 3 switch or router a “switch-on-a-stick” router. The traffic from one guest OS destined for a second guest OS on that platform traverses the physical NIC and the Ethernet twice.

The challenge that this vNIC technology presents to the network architecture is that you have a mix of virtual and physical devices in the infrastructure. Parts of the traditional access layer are effectively moved to the virtual realm. However, virtual NICs and virtual switches often do not have the same access controls, QoS capabilities, monitoring, and other features as access-level physical devices. Also, the virtual and physical elements might not be manageable from the same management platforms, which adds complexity to network management and troubleshooting. In any system architecture that employs virtual elements, enable efficient monitoring and management of both the physical and the virtual elements, and their points of intersection.

1.3 Characteristics of common virtualization environments

For a comprehensive discussion of available virtualization environments and their respective capabilities, see “IBM Data Center Networking: Planning for Virtualization and Cloud Computing,” SG24-7928-00. Table 1-1 summarizes those environments, their networking capabilities, and functions that must be implemented in the external network to support virtualization.

Table 1-1 Common virtualization environment characteristics

Environment	System (hypervisor) networking capabilities	VM mobility networking requirements
IBM System z®	IBM System z servers provide a wide range of interface options for connecting the system to a network. OSA-Express uses a direct memory access (DMA) protocol to transfer the data to and from the TCP/IP stack. The OSA-Express Ethernet features support IEEE standards 802.1p/q (priority tagging and VLAN identifier tagging). A IBM z/VM® virtual switch can operate at Layer 2 (data link layer) or Layer 3 (network layer), and bridges real hardware and virtualized LANs. By default, the virtual switch operates in IP mode (Layer 3) and data is transported in IP packets.	In the past, the preferred interconnectivity for IBM System z servers used TCP/IP (Layer 3) such as Virtual IP Addressing, OSPF “stub” areas, and dynamic routing protocols. IBM System z servers also support Layer 2 connectivity, which is useful for z/VM environments that run Linux KVM virtual machines.
IBM Power Systems™	IBM POWER® Hypervisor™ is the foundation for virtualization on IBM POWER System servers. It allows the hardware to be divided into multiple partitions, and ensuring isolation between them. IBM POWER Hypervisor provides IEEE VLAN compatible virtual switches, virtual Ethernet adapters, and IEEE 802.1q VLAN tagging. Live Partition Mobility makes it possible to move running IBM AIX® or Linux partitions from one physical IBM POWER6® and IBM POWER7® processor-based server to another without disruption.	Typically Layer 2, but also potentially Layer 3, including Virtual IP addressing and dynamic routing protocols much like IBM System z servers.

Environment	System (hypervisor) networking capabilities	VM mobility networking requirements
<p>IBM System x: VMware VMware is the current market leader in System x server virtualization. Their flagship product, vSphere, provides the infrastructure and management solutions for large enterprise level virtual environments. VMware was the first to offer migration technology (vMotion) to enable customers to quickly reprovision their data centers.</p>	<p>There are two types of virtual switches: vNetwork Standard Switch (vSS) and the vNetwork Distributed Switch (vDS). The vSS is configured at the ESX host level. The vDS is configured at the vCenter level and functions as a single virtual switch across the associated ESX hosts.</p>	<p>Layer 2 (“vMotion”) VMware provides the Software Development Kit (SDK) to third-party partners to create their own plug-in distributed virtual switches. Both the Nexus 1000V and the IBM Distributed Switch 5000V use that SDK to offer enhanced vSwitches that replace the VMware vDS.</p>
<p>IBM System x: Microsoft Hyper-V Microsoft Hyper-V (hypervisor) product runs on Windows 2008 Server and is provided with the server software. It supports Microsoft guest virtual machines and some Linux guest operating systems. With the backing of Microsoft, Hyper-V is gaining share in the server virtualization market.</p>	<p>At least one VLAN must be created to allow the VMs to communicate with the network. VLAN creation is supported by Hyper-V. The physical NIC is then configured to act as a virtual switch on the parent partition. At this time NIC teaming is not supported.</p>	<p>Layer 2</p>
<p>IBM System x: Xen Xen is an open source hypervisor that is available for Linux and Solaris operating systems. Citrix XenServer is a commercial and fully supported Xen hypervisor. Citrix also offers an accompanying suite of products for networking and management under the umbrella of Citrix Essentials. This suite can manage both Xen and Microsoft Hyper-V platforms.</p>	<p>Ethernet bridging allows VMs to communicate and share physical NICs.</p>	<p>Layer 2 (“XenMotion”)</p>
<p>IBM System x and IBM System z Kernel-based virtual machine (KVM) is a hypervisor that is rapidly gaining interest. KVM has been part of the Linux kernel since 2.6.20 and provides native virtualization on Intel VT and AMD-V processors. Currently the management tools required for data center KVM deployments are still limited, but they are evolving rapidly.</p>	<p>Uses User-Mode VirtIO drivers to emulate storage and network I/O (<i>paravirtualized I/O</i>). Supports SR-IOV adapters for hypervisor-bypass.</p>	<p>Layer 2</p>



Introducing VMready

This chapter introduces the IBM System Networking VMready technology. This chapter includes the following sections:

- ▶ Network support of system virtualization
- ▶ VMready overview
- ▶ VMready with Edge Virtual Bridging (IEEE 802.1Qbg)
- ▶ VMready without Edge Virtual Bridging

2.1 Network support of system virtualization

As noted in Chapter 1, “What you need to know about system virtualization” on page 3, system virtualization inevitably results in consolidating network traffic. This consolidation drives requirements for greater bandwidth to servers, which is a key driver behind rapid adoption of 10 Gb Ethernet.

Enforcing IT service-specific security policies and performance requirements such as virtual LANs, access control lists, and quality of service poses a difficult challenge for organizations using system virtualization. Virtual workloads are often all treated the same in terms of network performance and security settings. If necessary, organizations can group systems into pools that have common network settings. However, this approach reduces or eliminates one of the fundamental benefits of system virtualization: Running applications at different settings in the same host to achieve better overall system utilization. This configuration assumes that different IT services have utilization peaks at different times.

Ideally, an IT organization provides a *single pool* of processing capacity from which it can deliver *all IT services*. Better service performance, scalability, efficiency, and availability can be achieved by enabling IT services to share common servers. But to accomplish this, IT environments must support workloads that have different performance or security requirements on the same hosts. This environment in turn requires the network to have visibility into the virtual machines that run on each server. It needs access so that it can enforce unique network policies for each workload. IBM System Networking VMready makes this visibility possible, making the network “VM aware”. Figure 2-1 shows the values of a VMready network.

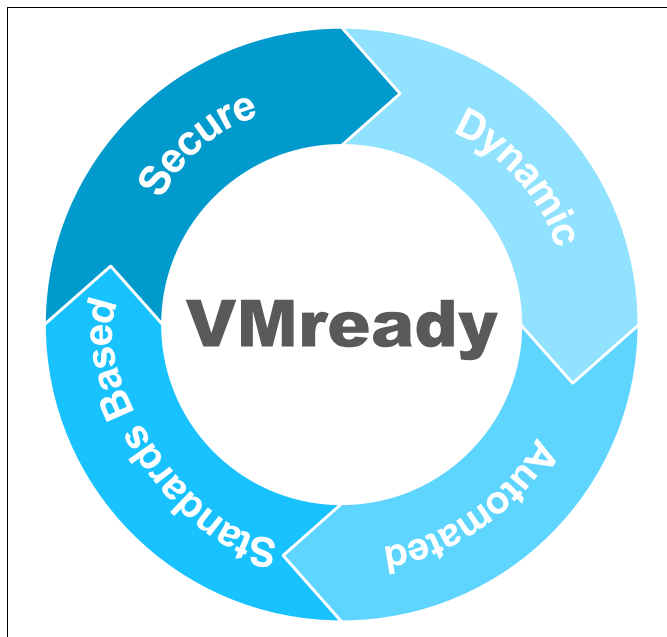


Figure 2-1 VMready value

An important capability of hypervisors is the ability to migrate running virtual machines between systems (called vMotion in VMware environments). Such dynamic virtual machine movement can be used to balance workloads or in response to a system failure. As virtual machines move within the environment, their unique network settings likewise need to be moved. With VMready, that capability is called IBM NMotion® (short for “network motion”).

Large organizations typically have separate groups for the management of servers and networks. Typically, the network team has responsibility for these activities:

- ▶ Enforcing security policies through the use of virtual LANs and access control lists (ACLs)
- ▶ Assuring appropriate network performance by using quality of service (QoS) settings

The use of hypervisor virtual switches drives the need for additional coordination between those groups. This need can increase the time required to deploy new IT services when most IT organizations are under significant pressure to reduce that time. For VMware environments, VMready addresses that problem by automating updates to VMware vSwitches as corresponding changes are made in the network. Furthermore, the new VMready Edge Virtual Bridging (IEEE 802.1Qbg) support provides a more-structured means of allowing the network team focus on delivering network services. The systems team likewise focuses on delivering system services. For more information, see 2.3.1, “VDP: Automating network switch configuration” on page 14.

2.2 VMready overview

In February 2009, BLADE Network Technologies (now an IBM company) delivered a unique solution, called VMready, to enable Ethernet switches to run these tasks:

- ▶ Discover virtual machines as they are started in the systems environment
- ▶ Enable network configurations (and even pre-provisioning) at a *virtual port* level rather than just at the switch physical port (Figure 2-2)

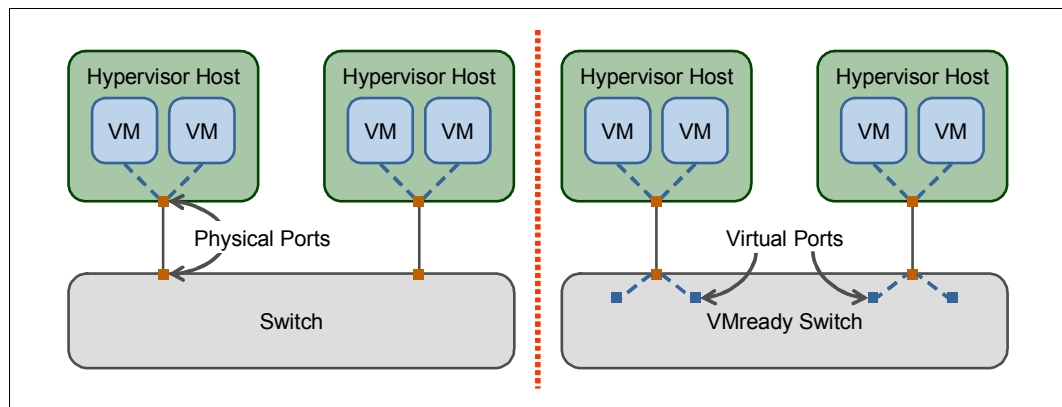


Figure 2-2 VMready versus non-VMready switches

- ▶ Track the migration of virtual machines across data centers and automatically reconfigure the network as the virtual machines migrate

These capabilities have these additional advantages:

- ▶ Come with the IBM Ethernet switch Networking Operating System at no additional charge
- ▶ Require no special hypervisor versions or additional software to be installed into the systems environment
- ▶ Work with all major hypervisors, including VMware, PowerVM, Hyper-V, Xen, KVM, and Oracle VM, without modification

VMready is now in its fourth generation:

- ▶ VMready 1.0 was released in February 2009 and introduced VM auto discovery, NMotion, and VM Groups for ease of configuration.
- ▶ VMready 2.0 was released in October 2009 and added VMware VI-API Integration, which enables simplified and consistent physical and virtual switch configuration.
- ▶ VMready 3.0, called VMready with Virtual Vision, was released in August 2010. It added a centralized database for managing VM Group configuration across the data center by using BLADE Harmony Manager.
- ▶ VMready 4.0 adds support for the newly available Edge Virtual Bridging (IEEE 802.1Qbg) standards.

Consideration: As of January 2012, the IEEE 802.1Qbg draft standard is in the final stages of IEEE approval. A fully ratified standard, based upon the current draft with minimal changes, is expected by mid-2012.

VMready support for the Edge Virtual Bridging (IEEE 802.1Qbg) standards works with any hypervisor that supports IEEE 802.1Qbg, including KVM. They also work with VMware that uses the IBM System Networking Distributed Switch 5000V for VMware. VMware does not yet natively support IEEE 802.1Qbg.

Restriction: An IBM Ethernet switch can be configured to support VMready Edge Virtual Bridging (IEEE 802.1Qbg) or the original VMready 3.0, but not both.

VMready virtual port configuration capabilities include:

- ▶ ACLs
- ▶ QoS attributes
- ▶ Virtual local area network (VLAN) membership
- ▶ Traffic shaping and monitoring

2.3 VMready with Edge Virtual Bridging (IEEE 802.1Qbg)

Although the standards terminology around Edge Virtual Bridging (IEEE 802.1Qbg) might seem daunting, the basic ideas are fairly straightforward.

2.3.1 VDP: Automating network switch configuration

One challenge that the standards seek to solve is having the appropriate network settings automatically defined in the network as virtual machines are started or migrated. This is important for virtualized environments because virtual machines are created and migrated much more frequently than physical systems.

The process for automating network switch configuration in support of system virtualization is illustrated in Figure 2-3.

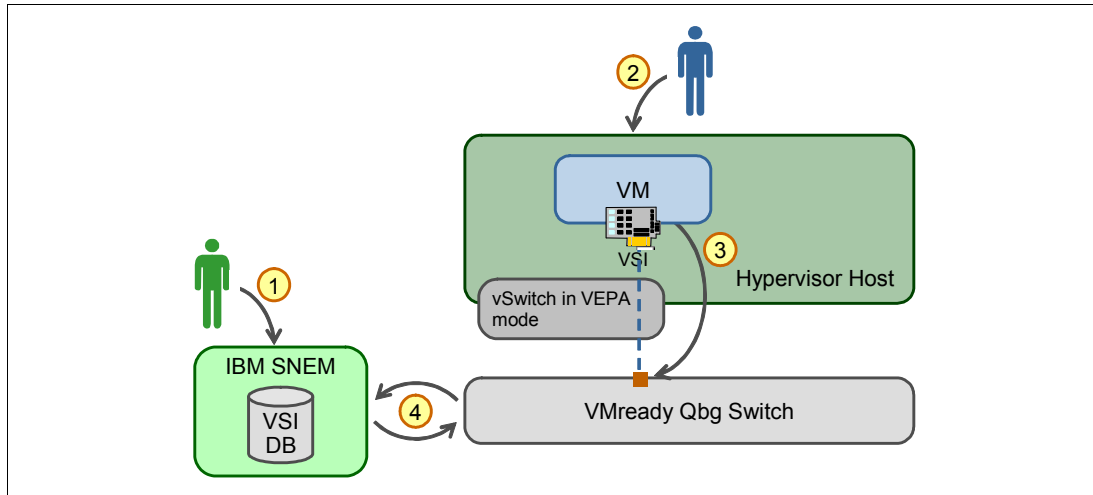


Figure 2-3 Automating network switch configuration

The main steps in the process include:

1. The networking team defines the supported portfolio of the VM port profiles and stores them, along with their corresponding network configuration parameters, in a database.
2. The systems team chooses the appropriate profile from that portfolio for the creation of a particular virtual machine.
3. The hypervisor sends a request to the network to associate the selected profile with the new VM.

Tip: The associate request is based upon the 802.1Qbg standard VSI Discovery and Configuration Protocol (VDP). The VDP associate request carries the MAC address of the new VM, the VLAN, and the VSI type that need to be used for that VM.

4. The adjacent network switch receives the request. It then queries the database of profiles to get the specific network configuration parameters, and applies those parameters to network traffic from that virtual machine.

Remember: The VDP associate process fails if one of these things happen:

- ▶ The requested VSI type is not found in the VSI type database
- ▶ The requested VLAN is not in the list of VLANs specified for that VSI type in the database

This process enables the network to always have the appropriate network settings available as virtual machines are migrated or new virtual machines are started.

Table 2-1 shows the standards terminology associated with a process and example products implementations.

Table 2-1 Standards terminology

Concept	Standards term	Implementation examples
Virtual Network Interface Card (vNIC)	Virtual Station Interface (VSI)	KVM, 5000V
vNIC port profile	VSI type	System Network Element Manager, 5000V Controller
Port profile database	VSI type manager	
Hypervisor request to associate a profile with a new VM	VDP	Hypervisor: KVM, 5000V Network: IBM BNT® 10 Gb ESM, G8264

2.3.2 VEPA: An alternative to vSwitches

Today, all major hypervisor environments provide virtual Ethernet switches (vSwitches) that facilitate physical network interface sharing, and enable local VM-to-VM switching. These vSwitches pose these challenges:

- ▶ They provide an additional level of “access switching” that is usually the responsibility of the server team. Network performance and security settings must then be configured for both physical switches and virtual switches. This process requires coordination between the server team and the network team. It is also more difficult to ensure that those settings are consistently enforced (particularly for locally switched VM-to-VM traffic).
- ▶ The vSwitches that are provided by the various hypervisors provide different performance and security capabilities. None of them offer capabilities that are as rich as those typically provided by physical Ethernet switches such as troubleshooting or collecting performance management data.

In general, place switch traffic as close to the virtual machines as possible. Ideally, place it in the vSwitch if the virtual machines are on the same host. Switching close to the communicating virtual machines minimizes traffic in the network and maximizes throughput.

However, in some situations, it might be better to send all traffic (including local VM-to-VM traffic) out of the physical network interface to be handled by the adjacent switch. This approach provides these advantages:

- ▶ Consolidating the responsibility for network performance and security to a single team, facilitating efficient and consistent policy enforcement.
- ▶ Taking advantage of the Edge Virtual Bridging support for automated network configuration. Doing so eliminates the need to manually configure individual physical switch ports and virtual switch port groups. For more information, see “VDP: Automating network switch configuration” on page 14.
- ▶ Using the powerful management and troubleshooting capabilities of the IBM System Networking Ethernet switches (such as the collection of sFlow data) for all traffic, including local VM-to-VM traffic.

The Edge Virtual Bridging (IEEE 802.1Qbg) standards include the Virtual Ethernet Port Aggregator option. This option causes *all traffic*, including local VM-to-VM traffic on the same host, to be sent out of the physical network interface to be handled by the adjacent switch.

Remember:

1. Non-virtualized systems have no need to send messages to themselves across an adjacent switch. Therefore, spanning tree protocols block such “hairpin turn” traffic. However, to support Virtual Ethernet Port Aggregator (VEPA), switches must be configured so traffic can be sent out of the same physical switch port that it came into. This support is called *reflective relay*.
2. The Link Layer Discovery Protocol (LLDP) is used for the exchange of capabilities, such as: reflective relay, VDP, and Edge Control Protocol (ECP). LLDP messages (called protocol data units, or PDUs) carry a list of capabilities in a simple type-length-value (TLV) format.
3. Current support of IEEE 802.1Qbg (KVM and the 5000V) is limited to VEPA only. Their vSwitches do not implement network policies based on those defined in a VSI type database.
4. Support for IEEE 802.1Qbg represents the next generation of VMready (VMready 4.0). However, one of the steps for implementing IEEE 802.1Qbg in IBM Ethernet switches is to disable VMready. This step just disables the previous (VMready 3.0) support because both cannot be used at the same time.

2.3.3 Edge Virtual Bridging (IEEE 802.1Qbg) for VMware

Today, VMware does not natively support the IEEE 802.1Qbg Edge Virtual Bridging standards. If you would like to take advantage of the IEEE 802.1Qbg open-standards-based automated network configuration or VEPA, use the IBM Distributed Switch 5000V for VMware (5000V).

As shown in Figure 2-4, the 5000V replaces the VMware native distributed vSwitch. It provides a managed distributed virtual switch with advanced networking capabilities for VMware vSphere 5.0. The 5000V is also the *only* product currently available that implements the Edge Virtual Bridging (IEEE 802.1Qbg) standards for VMware environments.

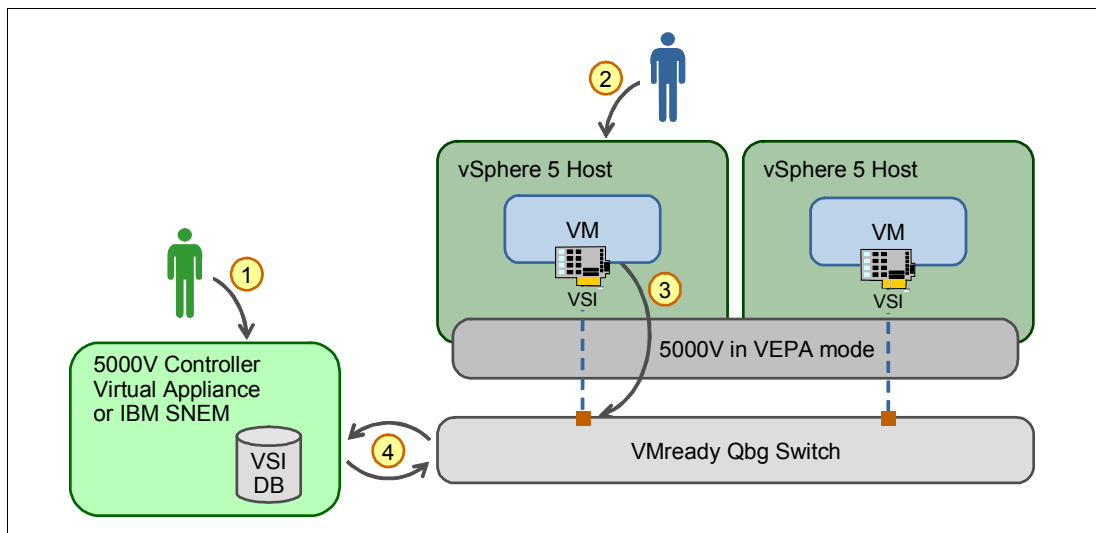


Figure 2-4 Edge Virtual Bridging (IEEE 802.1Qbg) for VMware

The basic steps of automating network switch configurations for VMware environments that use Edge Virtual Bridging (IEEE 802.1Qbg) are identical to those of any other hypervisor that

supports it. For more information, see 2.3.1, “VDP: Automating network switch configuration” on page 14.

The 5000V has these additional capabilities:

- ▶ Manageability: Telnet, SSH, SNMP, TACACS+, RADIUS, Industry Standard CLI
- ▶ Advanced networking features: L2-L4 ACLs, Static, and Dynamic port aggregation, Private VLAN (PVLAN), QoS, Edge Virtual Bridging (IEEE 802.1Qbg)
- ▶ Network troubleshooting: SPAN, ERSPAN, sFlow, Syslog, VM network statistics

2.4 VMready without Edge Virtual Bridging

It will take a while before Edge Virtual Bridging (IEEE 802.1Qbg) is available in all hypervisor environments. VMready provides similar automated configuration capabilities for environments that do not yet support Edge Virtual Bridging, but in a different way.

Without Edge Virtual Bridging, the hypervisor does not “announce” the addition of a new VM to the network. Therefore, the network must recognize the presence of a new VM when it first sends traffic. The network administrator must then assign the network port configuration for each new VM. This assignment can happen either when the VM first sends traffic or through pre-provisioning by using the Ethernet Media Access Control (MAC) address (or VM names in VMware environments). See Figure 2-5 for an overview of the process.

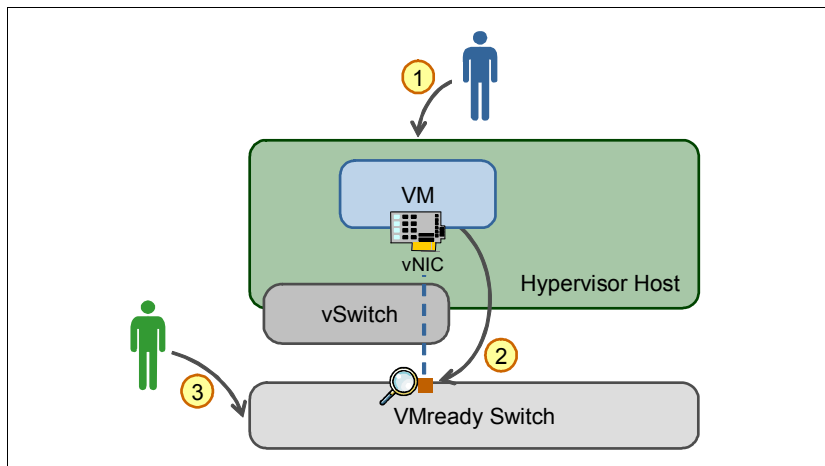


Figure 2-5 VMready without Edge Virtual Bridging (IEEE 802.1Qbg)

VMready support for virtualization without Edge Virtual Bridging can be summarized as follows:

1. When the systems administrator creates a VM, a corresponding vNIC and MAC address are created.
2. When the VM first sends traffic (or when it is pre-provisioned by a network administrator), VMready automatically creates a unique virtual port for it. Each virtual port is uniquely identified by its MAC address.

For most hypervisors, VMready uses the Organizationally Unique Identifier (OUI) from the MAC address to identify VMs. It examines the OUI in the MAC addresses on the downlink ports to identify a MAC that is known to be associated with a VM from a server virtualization vendor.

Table 2-2 shows the OUIs associated with various vendors.

Table 2-2 OUIs associated server Virtualization software

OUI (hex)	Vendor
00-0C-29	VMware
00-50-56	VMware
00-16-3E	XenSource
00-03-FF	Microsoft
00-0F-4B	Virtual Iron
00-18-51	Parallels

Other hypervisors, such as KVM and PowerVM, use locally administered MAC addresses for their VMs. These local addresses are identified by a “1” in the seventh bit of the MAC address. VMready does not, by default, interpret locally administered MAC addresses as VMs. However, VMready can be directed to do so with the command:

```
virt vmrmisc lmac
```

The MAC is then put in a Port/VM table that carries the association of the VM with its original port. When the VM moves to a new port, it is discovered and the Port/VM table is updated to reflect the new location.

3. The network administrator configures the new virtual port with networking parameters such as VLANs, ACLs, and QoS as though it was a physical port. For VMware environments, the configuration can be automatically exported to the hypervisor for vSwitch configuration as well.

Tip: A significant advantage of VMready Edge Virtual Bridging (IEEE 802.1Qbg) is that it eliminates the need for step 3. The network interface is configured as each new VM is brought up.

Thereafter, VMready tracks virtual machines in real time as they migrate, and automatically moves the virtual port along with its network configurations to the new physical location. This process ensures that VMs are always correctly configured in the network no matter where they move. It also eliminates the need for manual reconfiguration of network ports when moving VMs.

VMready allows for a “define once, use many” configuration. After the initial network configurations, VMs can move as events require, retaining their configurations from origin to destination. This process reduces the administrative burden of performing repetitive configurations and reduces the risk of misconfiguration.

2.4.1 Virtual machine identities

The foundation of VMready is the ability to identify and then monitor virtual machines within the physical hosts attached to the switch. By identifying each VM uniquely, VMready enables the administrator to apply the same philosophy to virtual machines as used with physical servers. VLAN and quality of service settings can be provisioned. Access Control List attributes can be set at a VM level with permit and deny actions based on Layer 2 to Layer 4 information.

For example, with VMready an organization can block all non-encrypted traffic (such as HTTP or telnet) through the use of a single ACL. This ACL must be applied both to physical servers and virtual machines.

Virtual machines can be configured in VMready by using a web GUI, a command-line interface (CLI), or by using IBM System Networking Element Manager.

2.4.2 Grouping

VMready provides a simple and intuitive way to group similar virtual machines together and assign the same networking policies to members of those groups. This configuration ensures consistent allocation of resources and security measures to meet service-level goals. VMready supports up to 32 groups. Members of the group retain the group attributes wherever they are located within the virtual environment.

Grouping significantly simplifies the administration tasks when managing large numbers of virtual machines because new virtual machines can be added to existing groups.

In BLADEOS 6.1 and later releases, VMready supports two types of groups:

- ▶ *Local VM groups* maintain VM configurations locally on the VMready switch (or stack of switches)
- ▶ *Distributed VM groups* are synchronized with VMware vCenter and enable consistent configuration of VMware vSwitches

Local VM Groups

Local VM group configurations are maintained on the switch, and do not synchronize with hypervisors. Local VM groups can include elements such as local switch ports and VMs that are connected to one of the switch ports, or are pre-provisioned on the switch.

As VMs move to different hosts, the configuration of their group identity and features moves with them. For example, VM Groups can be used to enforce external switch VLAN tagging.

Distributed VM Groups

In BLADEOS 6.1 and later, Distributed VM Group configurations are automatically synchronized with the VMware vCenter. This synchronization creates Port Groups with the same network configurations on all required vSwitches. Consistent network policies are enforced both in the physical switches and in the VMware vSwitches.

VMready provides VMware users with a single-pane-of-glass network management console for virtual machines. As new virtual machines are added to a distributed group or a group configuration is changed, VMready automatically updates the VMware vCenter. This process distributes the changes to all associated hypervisor vSwitches. This integration simplifies administrative tasks and reduces the chance of configuration error.

Restriction: At this time, the distributed VM group synchronization with VMware works only with vSphere Standard Switches, not with vSphere Distributed Switches.

2.4.3 NMotion

VMready NMotion (network motion) is the ability to detect VM migrations as they occur. The system then moves the associated set of network attributes (ACL, QoS, and VLAN) among downlink ports on the switch. VM migrations are detected by capturing Reverse Address Resolution Protocol (RARP) or some other network discovery traffic originated by the

hypervisor on behalf of a VM. The hypervisor generates administrative traffic on behalf of the moving VM to run route setup before the VM starts sending traffic. For VMware, this process is run at around 90% VMotion complete.

Trying to accomplish similar network reconfiguration through manual procedures without VMready is impractical in large virtual infrastructures. You can manually duplicate the configuration on the destination port before the migration if you knew that the VM was about to move. However, such a process is labor-intensive and the risk of error is high. Alternatively, you can try to provision for all possible combinations of VM port configurations (VLANs) on every port in the network. However, this approach does not scale well for large environments and might actually defeat the purpose of configurations such as using VLANs for security.

VMready solves those network configuration management issues through NMotion, which manages the distribution of network policies in real time when VMs migrate. NMotion can identify and then track virtual machines as they move. This process guarantees that the underlying services not only remain available, but also retain their network characteristics of ACLs, QoS, and VLAN membership.



Management for VMready

This chapter focuses on management for VMready environments. This chapter includes the following sections:

- ▶ IBM System Networking Element Manager Overview
- ▶ Configuring VMready with System Networking Element Manager
- ▶ Monitoring and troubleshooting your VMready installation

3.1 IBM System Networking Element Manager Overview

The IBM System Networking Element Manager is a *virtual appliance* packaging of tools that help manage IBM System Networking hardware offerings. The System Networking Element Manager virtual appliance is a pre-integrated virtual machine image that installs and delivers value quickly. The IBM System Networking Element Manager package consists of these applications:

- ▶ IBM System Networking Element Manager component, which provides monitoring, configuration management, and reporting capabilities for IBM System Networking blade and rack switches.
- ▶ IBM Tivoli® OMNibus and Network Manager, which provides real-time discovery, monitoring and event management for network environments.
- ▶ Tivoli Netcool® Configuration Manager, which automates network configuration management tasks, controls network device access, and ensures network policy compliance.

Each of these components is addressed in further detail in the following sections.

3.1.1 IBM System Networking Element Manager component

The IBM System Networking Element Manager component (formerly BLADEHarmony Manager) was developed specifically for managing the IBM System Networking portfolio of blade and rack switches. It helps network administrators to monitor, configure, and view summary reports for IBM switches.

The IBM System Networking Element Manager component has the following highlights:

- ▶ Global view of all IBM System Networking top of rack and embedded switches with simplified asset and fault management
- ▶ Centralized point of administration to improve service delivery while reducing management costs
- ▶ Remote monitoring and switch management
- ▶ Multi-vendor virtual machine network configuration to support secure migration of virtual machines across the data center (see “VMready Across the Data Center” on page 39)

For more information about the IBM System Networking Element Manager component, see:

http://www-03.ibm.com/systems/networking/software/blade_harmony/

3.1.2 IBM Tivoli OMNibus and Network Manager

IBM Tivoli OMNibus and Network Manager helps network operations with comprehensive multivendor network discovery, real-time monitoring and event automation, and network-topology-based root cause analysis. It helps direct operations teams find the most likely “root cause” when network outages occur.

As a component of the IBM System Network Element Manager package, IBM Tivoli OMNibus and Network Manager is intended to be used for managing IBM System Networking equipment. Additional licensing is needed if you want to use this component for networking equipment not purchased from IBM System Networking.

IBM Tivoli OMNibus and Network Manager has the following highlights:

- ▶ Delivers a central point of management for all IT and network operations
- ▶ Optimizes service availability, reliability, and asset utilization through automated event correlation, isolation, and resolution capabilities that allow rapid identification and resolution of the most critical problems
- ▶ Provides real-time web dashboard views with customizable displays for events, service views, and operational indicators for consolidated operations management
- ▶ Automates event deduplication and correlation, root-cause fault isolation, and in-context prioritization and diagnosis to increase productivity and operational efficiency
- ▶ Discovers devices and the physical port-to-port connectivity, as well as the logical topology including VLANs, to deliver highly accurate, real-time information about network availability

For more information about IBM Tivoli OMNibus and Network Manager, see:

<http://www-01.ibm.com/software/tivoli/products/omnibus-network-mgr/>

3.1.3 IBM Tivoli Netcool Configuration Manager

IBM Tivoli Netcool Configuration Manager helps network administrators with multivendor network configuration management, automating network configuration changes with the highest levels of standardization, accuracy, and control.

As a component of the IBM System Network Element Manager package, IBM Tivoli Netcool Configuration Manager is intended to be used for managing IBM System Networking equipment. Additional licensing is needed if you want to use this component for networking equipment not purchased from IBM System Networking.

IBM Tivoli Netcool Configuration Manager has the following highlights:

- ▶ Intuitive user interface that allows network device configuration that supports users ranging from expert network engineers to novice operators.
- ▶ Manages the complete device configuration lifecycle: Baseline, search, configure, test, approve, and track.
- ▶ Can limit user access to part of a device configuration or selected commands. This feature is essential for virtualized or multi-service environments, or where there is a need for greater device control.
- ▶ Provides a reusable policy framework for configuration policy validations of regulatory mandates, security, and operational policies such as engineering standards.

For more information about IBM Tivoli Netcool Configuration Manager, see:

<http://www-01.ibm.com/software/tivoli/products/netcool-configuration-manager/>

3.2 Configuring VMready with System Networking Element Manager

Most VMready configuration can be done directly with the switch by using either the command-line interface (CLI) or the web-based graphical user interface. For more information, see Part 2, “Implementing a VM-aware network” on page 43. This chapter is focused on configuring VMready with the System Networking Element Manager.

VMready for Edge Virtual Bridging (IEEE 802.1Qbg) is different from VMready when not using 802.1Qbg. Their respective configurations are likewise different.

3.2.1 Configuring VMready for Edge Virtual Bridging (IEEE 802.1Qbg)

For an overview of how VMready for Edge Virtual Bridging (IEEE 802.1Qbg) works, see 2.3, “VMready with Edge Virtual Bridging (IEEE 802.1Qbg)” on page 14.

To configure VMready for Edge Virtual Bridging (IEEE 802.1Qbg), perform these steps.

Step 1: Create an Edge Virtual Bridging profile

In the Edge Virtual Bridging (EVB) profile, enable Reflective Relay and the VSI Discovery and Configuration Protocol (VDP) as shown in Figure 3-1. Reflective Relay allows the switch to send messages back out the same physical port that they came into. This configuration supports sending same-host VM-to-VM traffic through the external switch. The VDP enables automatic switch configuration as new VMs are started or VM are migrated.

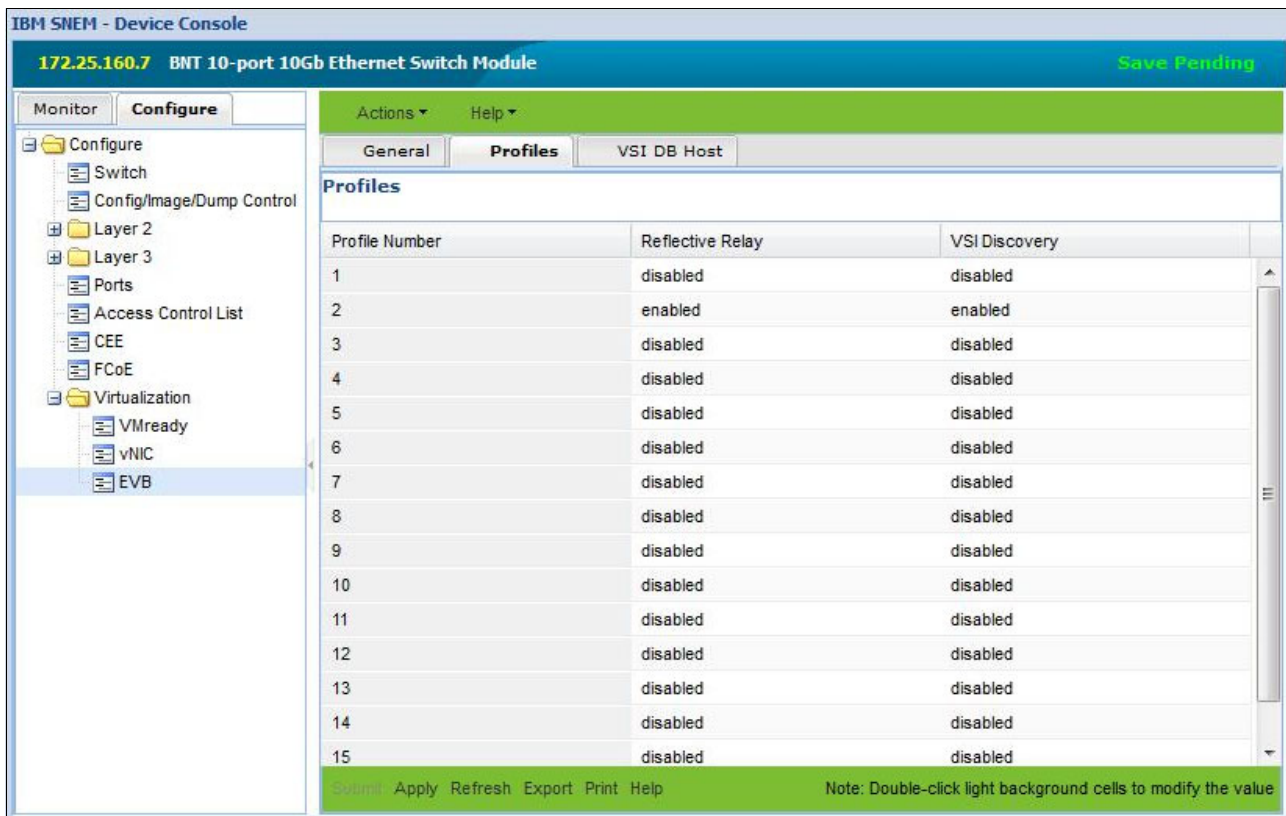


Figure 3-1 EVB Profile

Step 2: Add the Edge Virtual Bridging profile to the port

This step associates the EVB profile configured in step 1 with specific physical ports. Associate EVB profiles with server ports as shown in Figure 3-2.

IBM SNEM - Device Console
172.25.160.7 BNT 10-port 10Gb Ethernet Switch Module Save Pending

Monitor **Configure**

Actions Help

Ports Threshold Rate Gigabit Link UDLD OAM ACL/QOS STP Port Priority

DHCP Snooping

Ports

Port	Name	State	VLAN Tag State	Default VLAN	PVID Tag State	Link Trap	BPDU Guard	DSCP Rem...	RMON	FDB Lear...	Flood Block...	Error Disable	EVB Profile
INT1	INT1	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT2	INT2	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT3	INT3	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT4	INT4	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT5	INT5	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT6	INT6	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT7	INT7	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT8	INT8	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT9	INT9	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT10	INT10	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT11	INT11	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT12	INT12	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0
INT13	INT13	enabl...	tagged	1	unta...	enabl...	disab...	disab...	off	enabl...	disab...	enabl...	0

Save Apply Refresh Export Print Help Note: Double-click light background cells to modify the value

Figure 3-2 Adding EVB profile to a port

Step 3: Configure ECP retransmission interval

The Edge Control Protocol (ECP) provides reliable delivery of service dispatcher units between the station and bridge, and between the port extender and bridge (Figure 3-3).

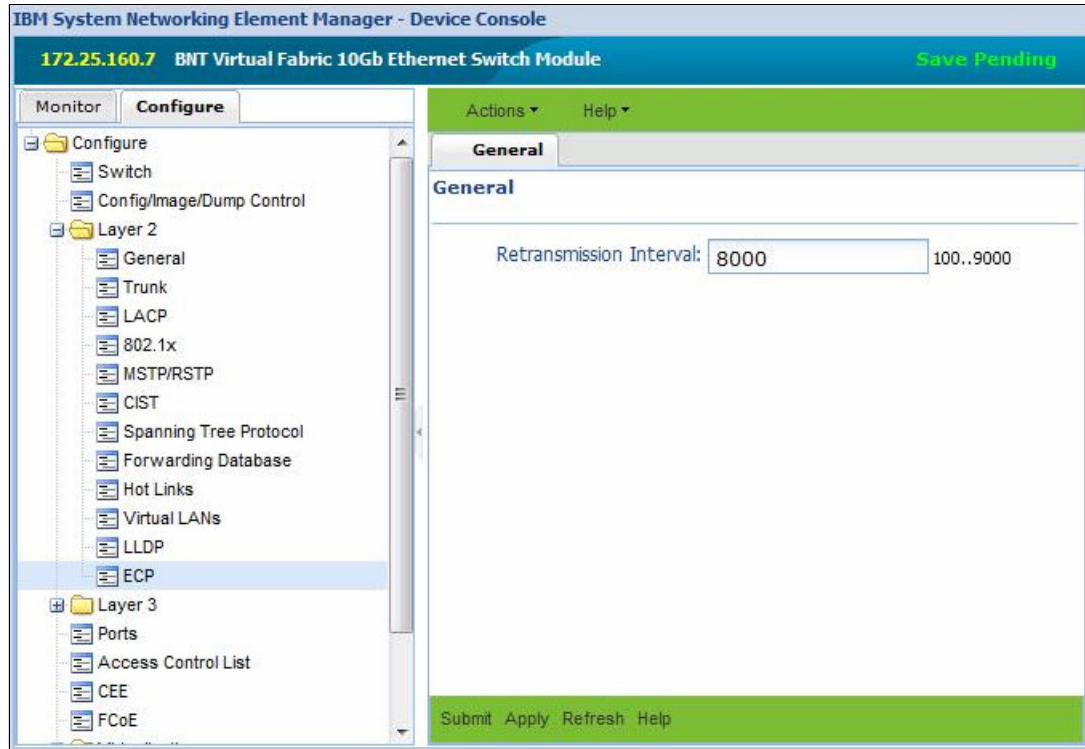


Figure 3-3 Configure ECP retransmission interval (8 seconds)

Step 4: Define supported VSI types and setup VSI type database

VMready for Edge Virtual Bridging (IEEE 802.1Qbg) queries a VSI type database for VSI Type Definition settings. Switch uses HTTP to connect to the VSI DataBase and retrieve definitions in XML format.

Either IBM System Networking Element Manager or the IBM Distributed Switch 5000V can act as the VSI type database.

Using IBM System Networking Element Manager as the VSI type database

This step directs the switch to reference the IBM System Networking Element Manager VSI type database (Figure 3-4).

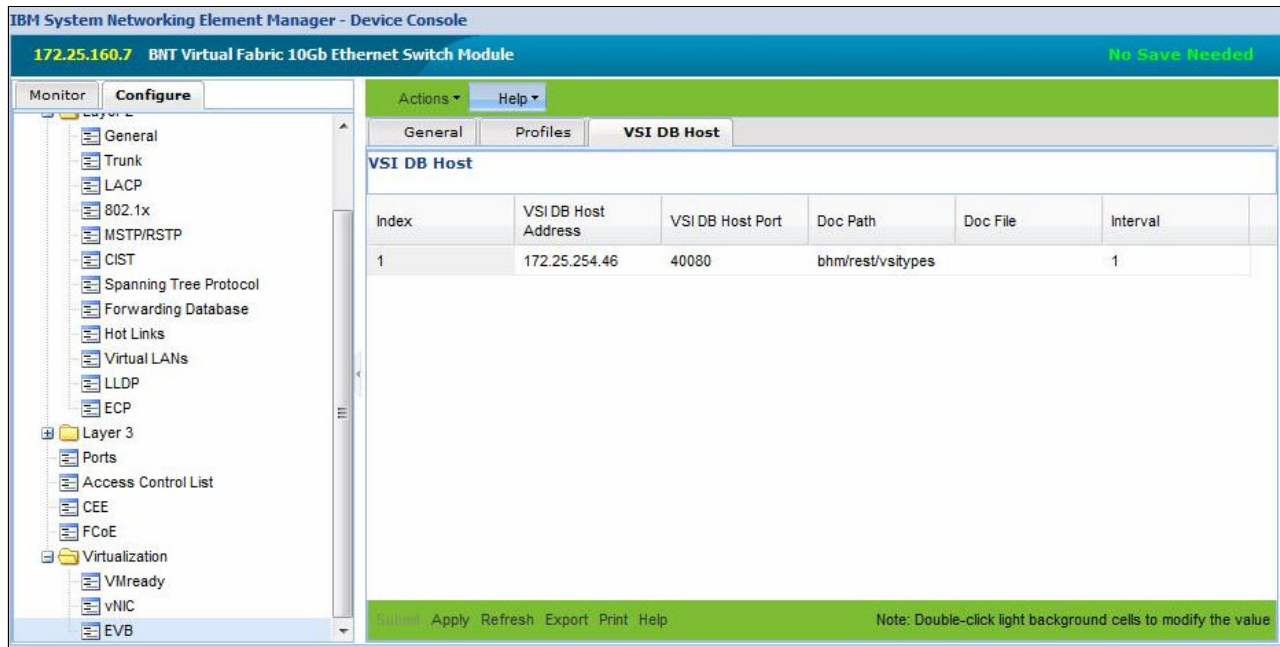


Figure 3-4 Using IBM System Networking Element Manager as the VSI type database

Tip: This configuration results in a database update request from the switch at:
<http://172.25.254.46:40080/bhm/rest/vsitypes>

Using the IBM Distributed Switch 5000V as the VSI type database

This step directs the switch to reference the 5000V VSI type database as shown in Figure 3-5.

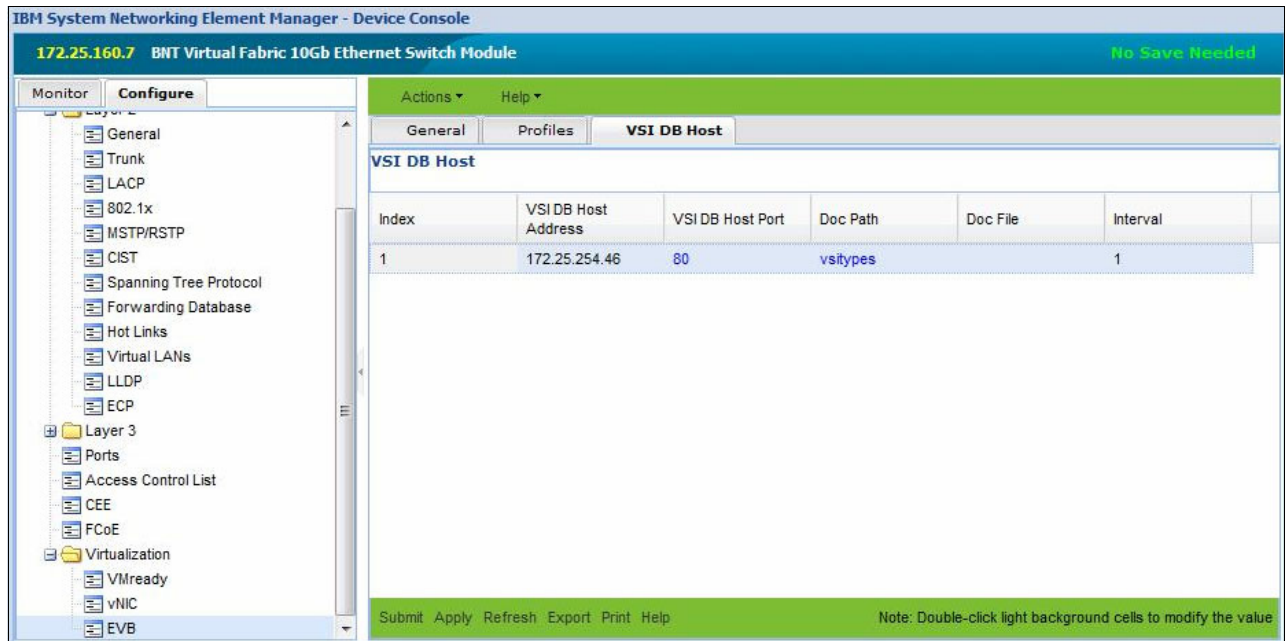


Figure 3-5 Using the IBM Distributed Switch 5000V as the VSI type database

Tip: This configuration results in a database update request from the switch at:
<http://172.25.254.46:80/vsitypes>

Step 5: Updating VSI type definitions

Add the supported VSI types as shown in Figure 3-6.

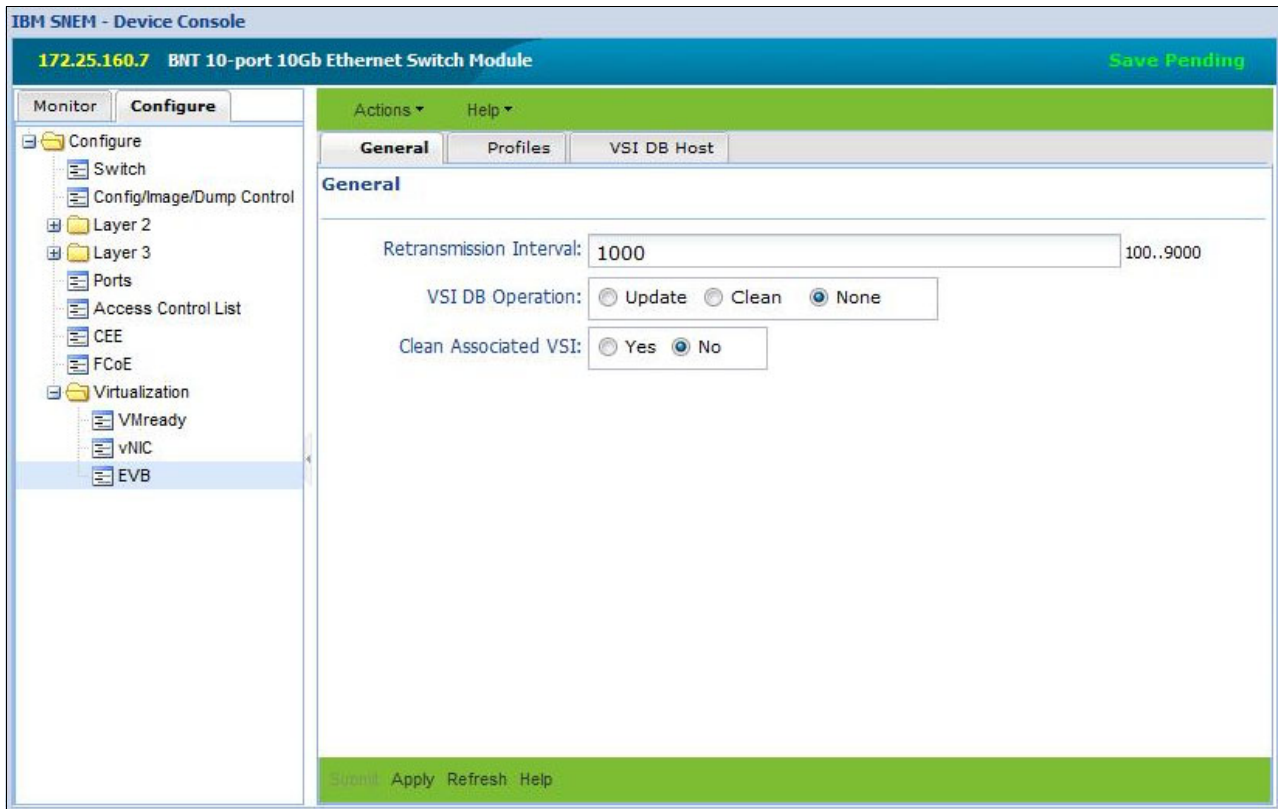


Figure 3-6 Updating VSI type definitions

Step 6: Enable Link Layer Discovery Protocol (LLDP)

Edge Virtual Bridging peers (switches and hypervisors) use Link Layer Discovery Protocol (LLDP) to exchange capabilities with respect to supporting 802.1Qbg (Figure 3-7). For example, LLDP is used to support reflective relay.

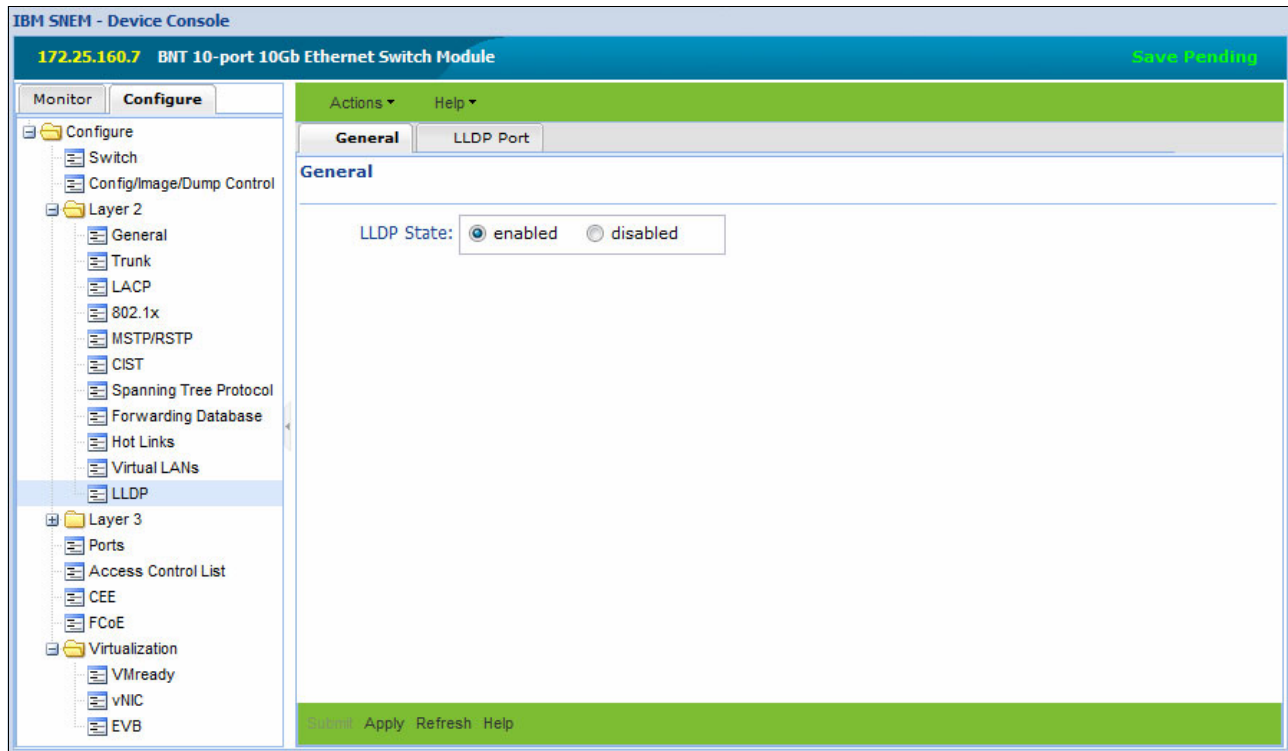


Figure 3-7 Enabling LLDP

Step 7: Disable VMready

Because you are using IEEE 802.1Qbg, disable the prior VMready support as shown in Figure 3-8.

Tip: Support for IEEE 802.1Qbg represents the next generation of VMready (VMready 4.0). However, one of the steps for implementing IEEE 802.1Qbg in IBM Ethernet switches is to “disable VMready”. This step just disables the previous (VMready 3.0) support because both cannot be used at the same time.

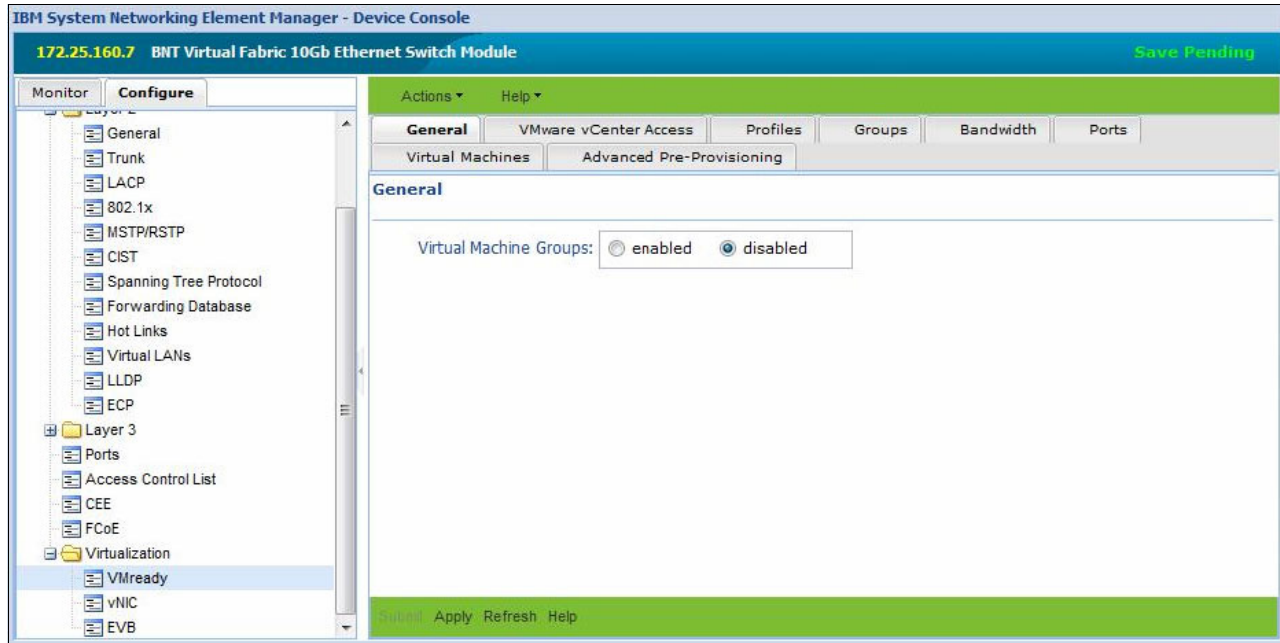


Figure 3-8 Disabling VMready

Remember: VMready is disabled by default. IEEE 802.1Qbg does not use any code from VMready, which therefore should not be enabled.

3.2.2 Configuring VMready when not using Edge Virtual Bridging

For an overview of how VMready works when not using Edge Virtual Bridging, see 2.4, “VMready without Edge Virtual Bridging” on page 18.

To configure VMready when not using Edge Virtual Bridging, perform the following steps,

Step 1: Enabling VMready

The global switch command to enable VMready is `virt enable` as shown in Figure 3-9.

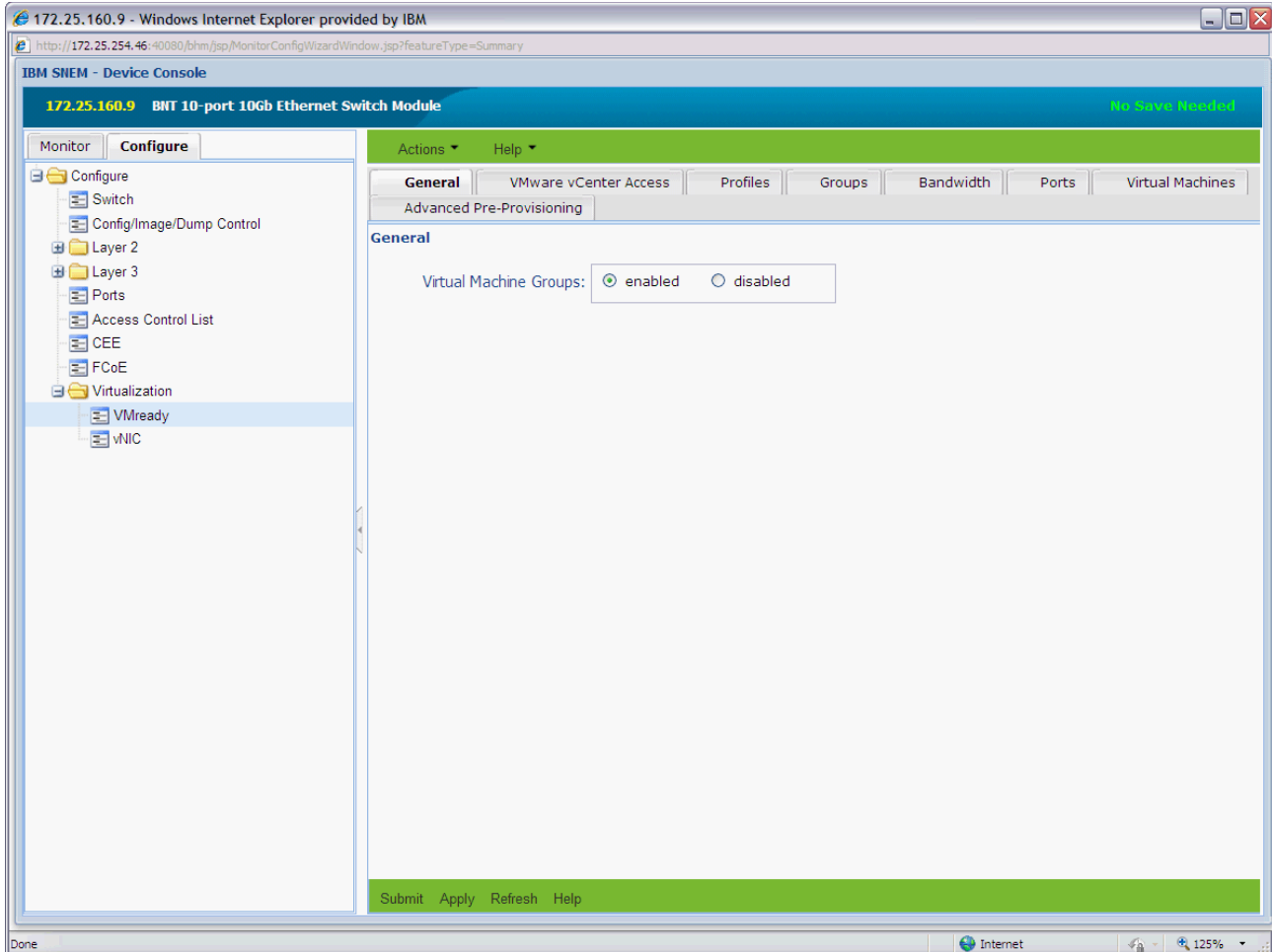


Figure 3-9 Enable VMready

Step 2: Define vmpfiles

The vmpfiles provide port group definitions for the VMs as shown in Figure 3-10.

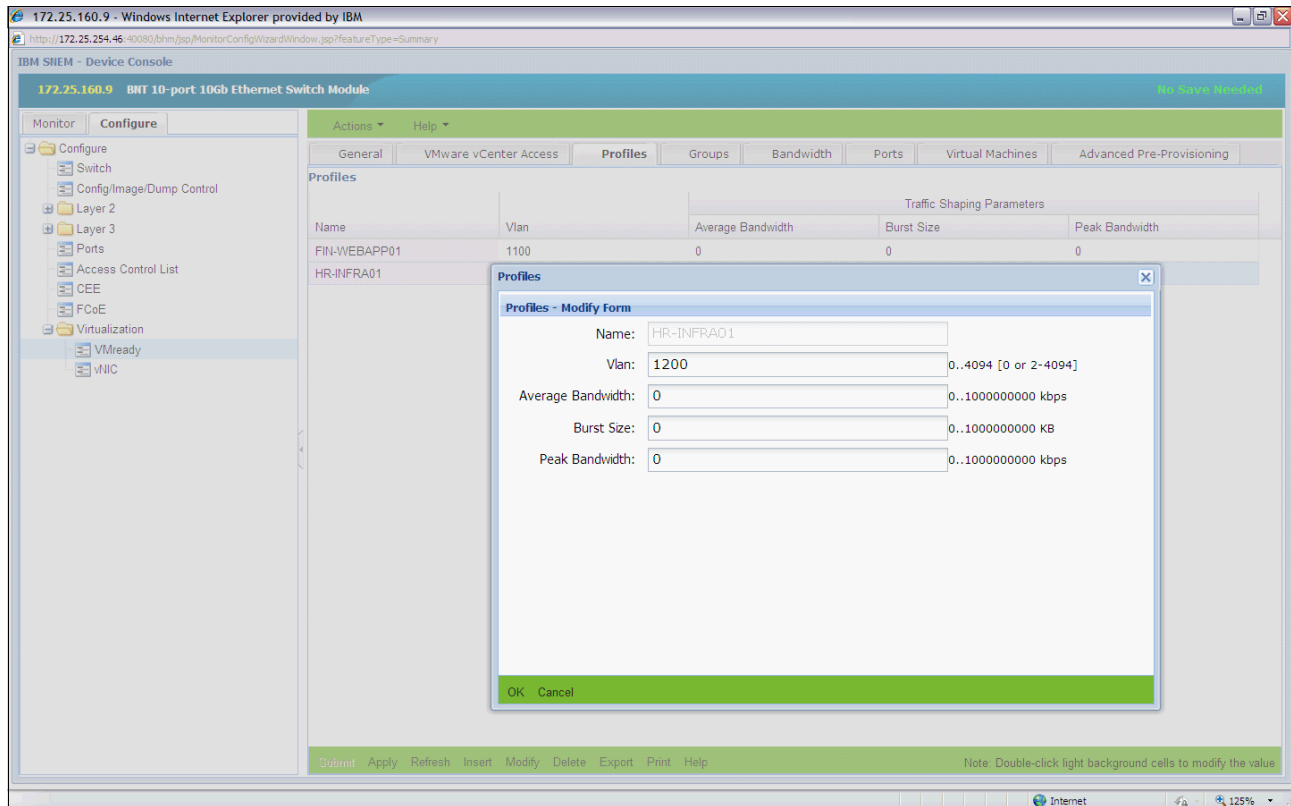


Figure 3-10 Define vmpfiles

Tip: Although vmpfiles are used in this example, you do not have to define vmpfiles for VMready to work. VLAN IDs can be directly configured on vmgroups instead of vmpfiles.

Step 3: Define group ACLs (vmaps)

Define a single group ACL (vmap) by using an available vmap number: 11. Define it to deny all traffic destined for port 80 as shown in Figure 3-11.

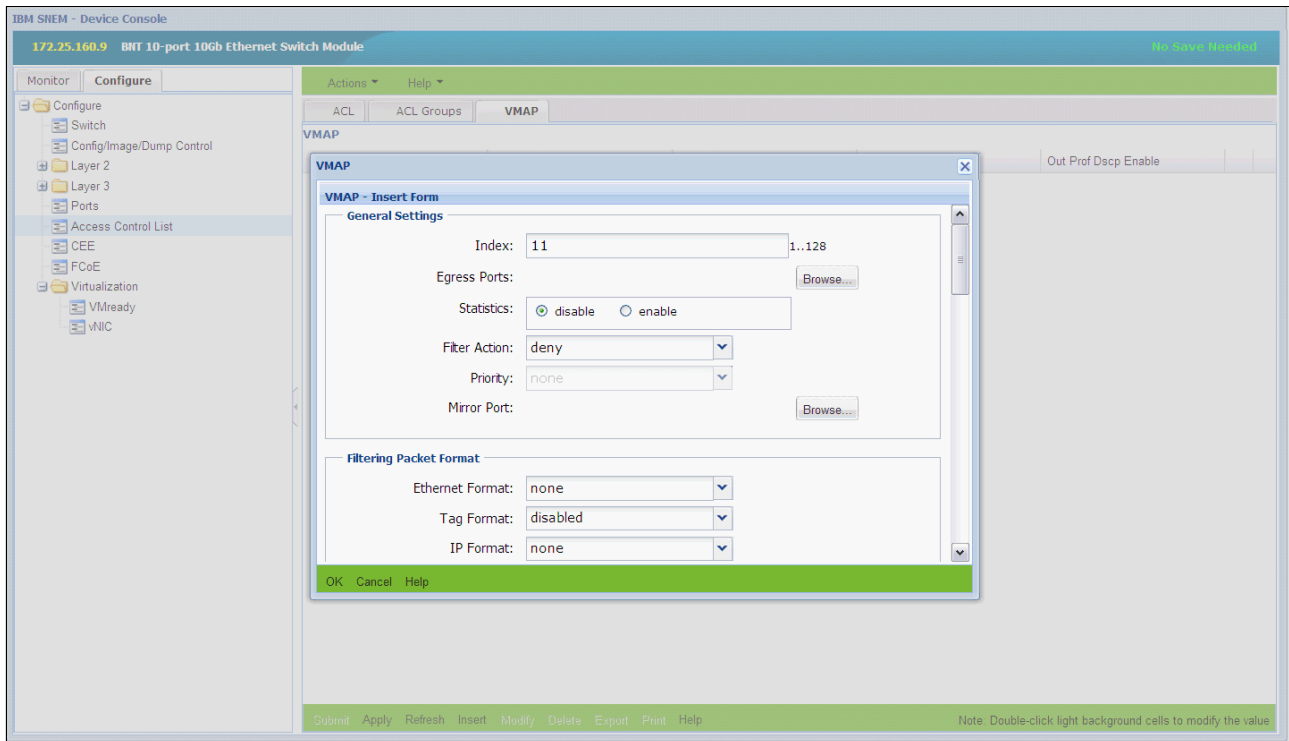


Figure 3-11 Define group ACLs

Step 4: Apply vmprofiles to vmgroups

Associate each **vmprofile** to a unique **vmgroup**. The **vmgroup** takes over the identity of the VLAN ID in the associated vmprofile to create a VM-Aware VLAN (Figure 3-12).

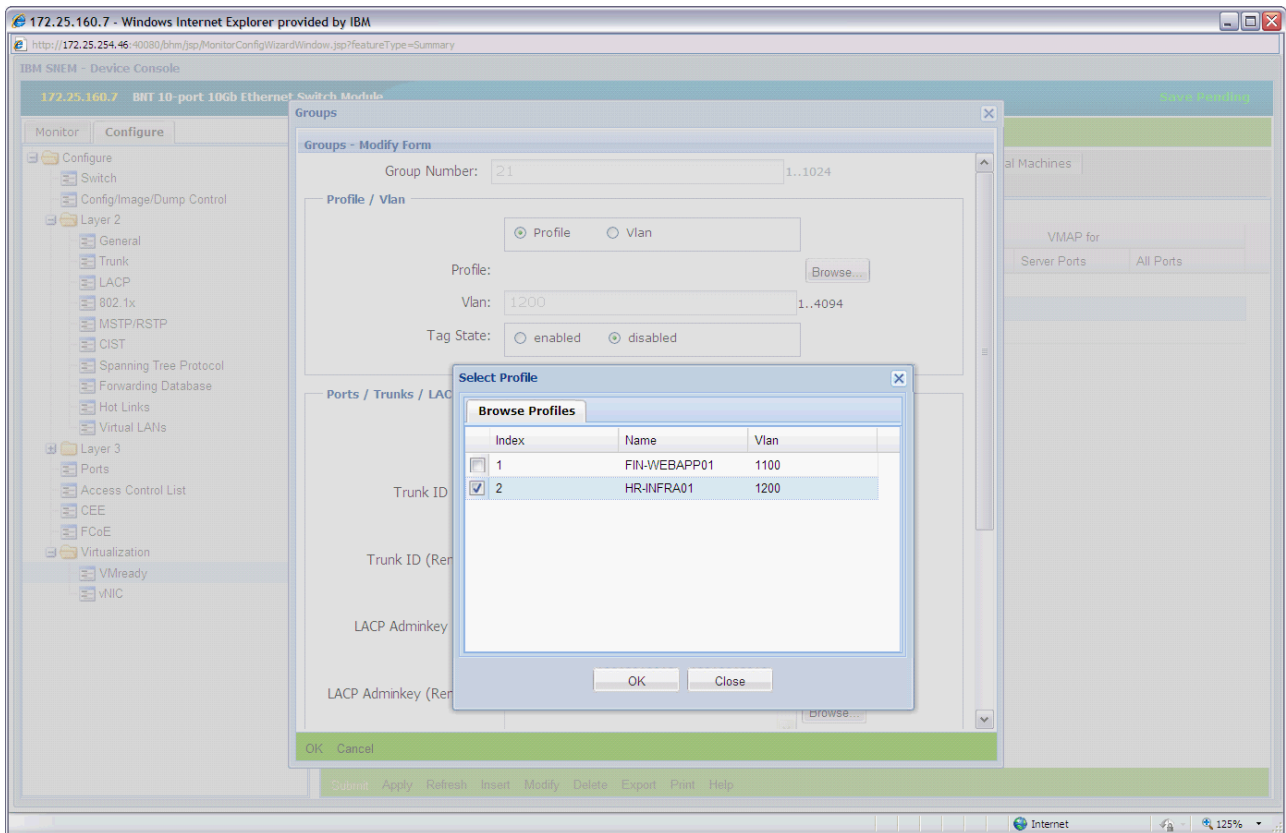


Figure 3-12 Apply profiles to vmgroups

Step 5: Apply vmaps to vmgroups

Allocate the vmap defined earlier to the group that contains the finance web application servers, WEB and DB, as shown in Figure 3-13.

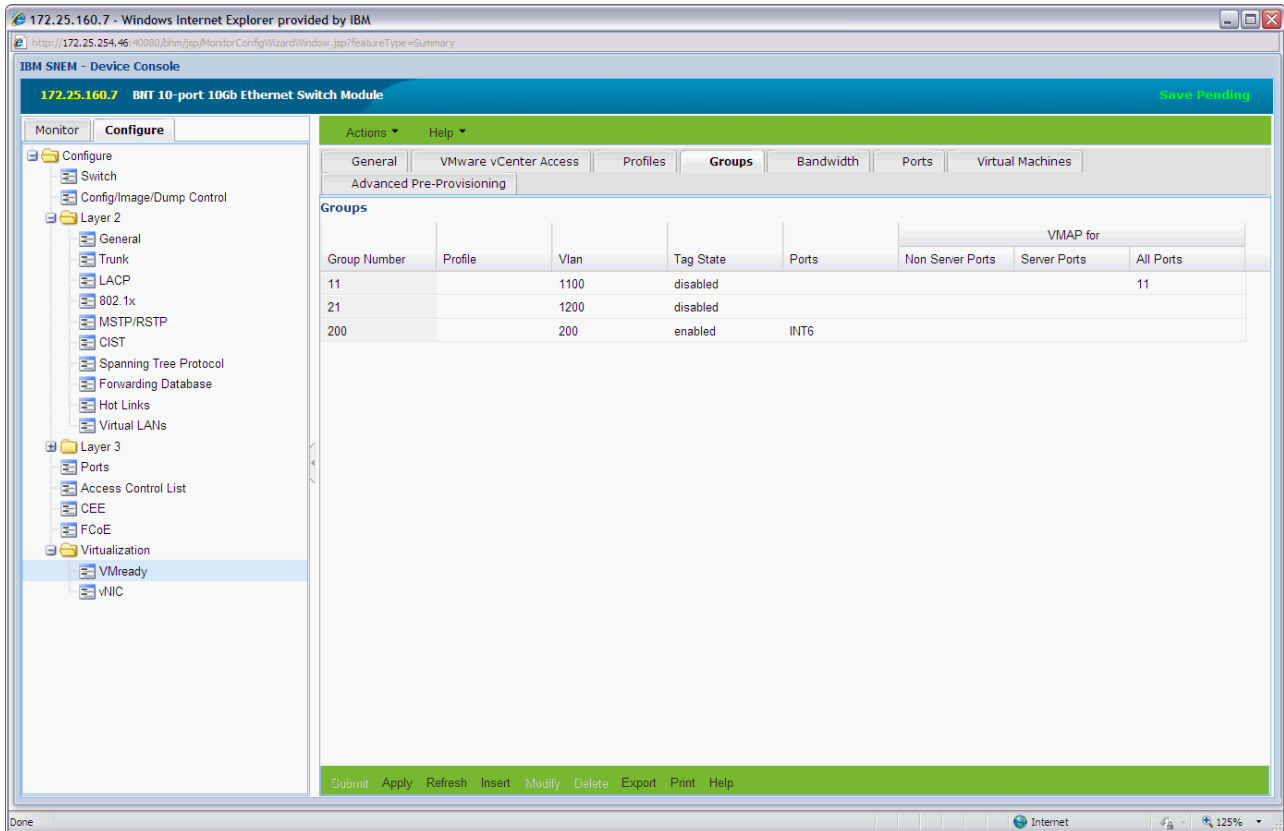


Figure 3-13 Apply vmaps to vmgroups

Step 6: Add VMs to vmgroups

Add VMs to your vmgroups. VMs are added based on the MAC address. For VMware, you can also use VM name, IP address, or UUID. If a VM has multiple virtual network adapters, specify the exact target MAC address as shown in Figure 3-14.

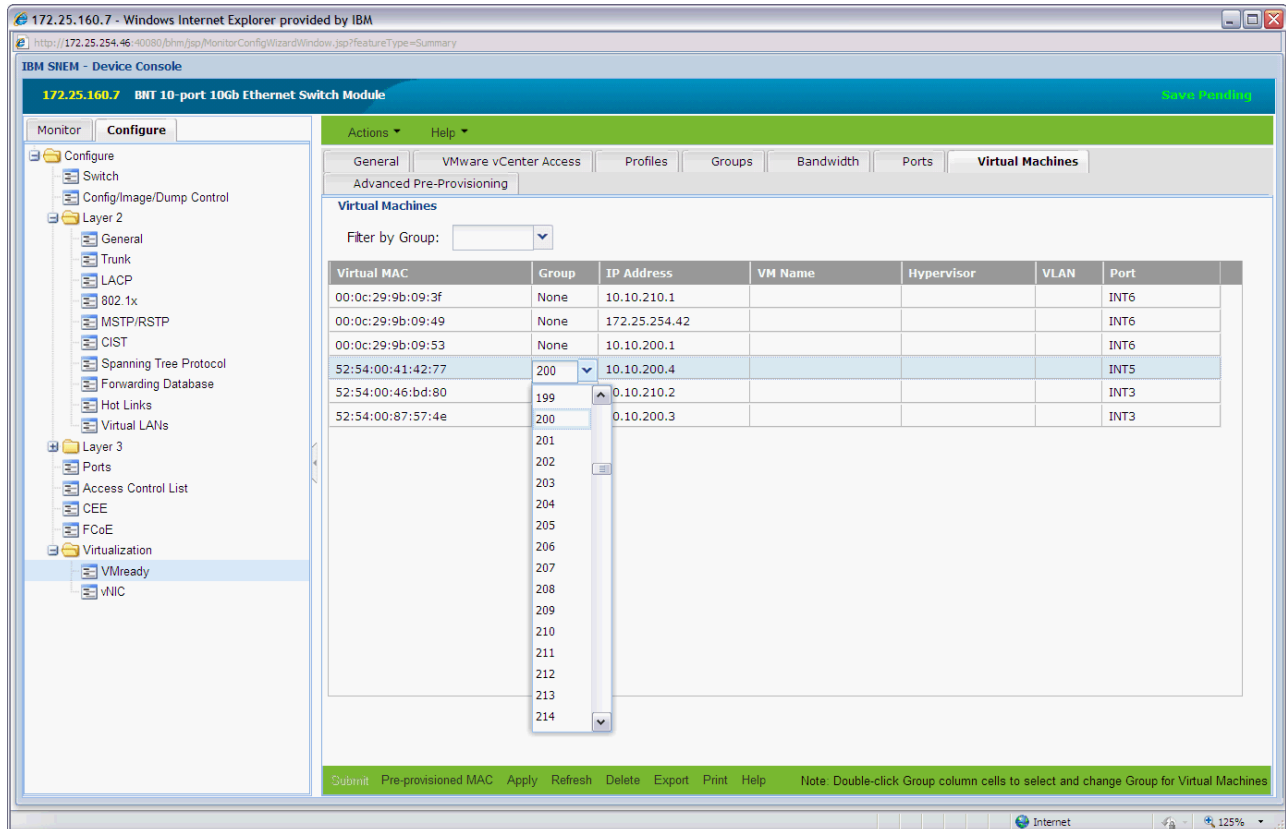


Figure 3-14 Add VMs to vmgroups

Step 7: Assign per-VM bandwidth policy

Assign a per-VM bandwidth shaping policy to ensure that the HR FILE server does not use all bandwidth on a host. Apply it for traffic that comes from the VM (txrate) and going to the VM (rxrate). Set the bandwidth limit at 1 Mbps.

Implement per VM bandwidth control by using the commands shown in Example 3-1

Example 3-1 Commands to implement per VM bandwidth control

```
Router(config)#virt vmpolicy vmbwidth FILE rxrate 1024 1024
Router(config)#virt vmpolicy vmbwidth FILE txrate 1024 1024
Router(config)#virt vmpolicy vmbwidth FILE bwctrl
```

VMready Across the Data Center

The network configuration of VMready groups can be migrated across switches and data centers by using IBM Systems Networking Network Element Manager. This capability, called VMready Across the Data Center, increases the scale of VMready beyond a single switch (or stack of switches). You can centrally define policies and distribute them to switches and, in the case of VMware, hypervisors.

Figure 3-15 shows an overview of how VMready with Virtual Vision works.

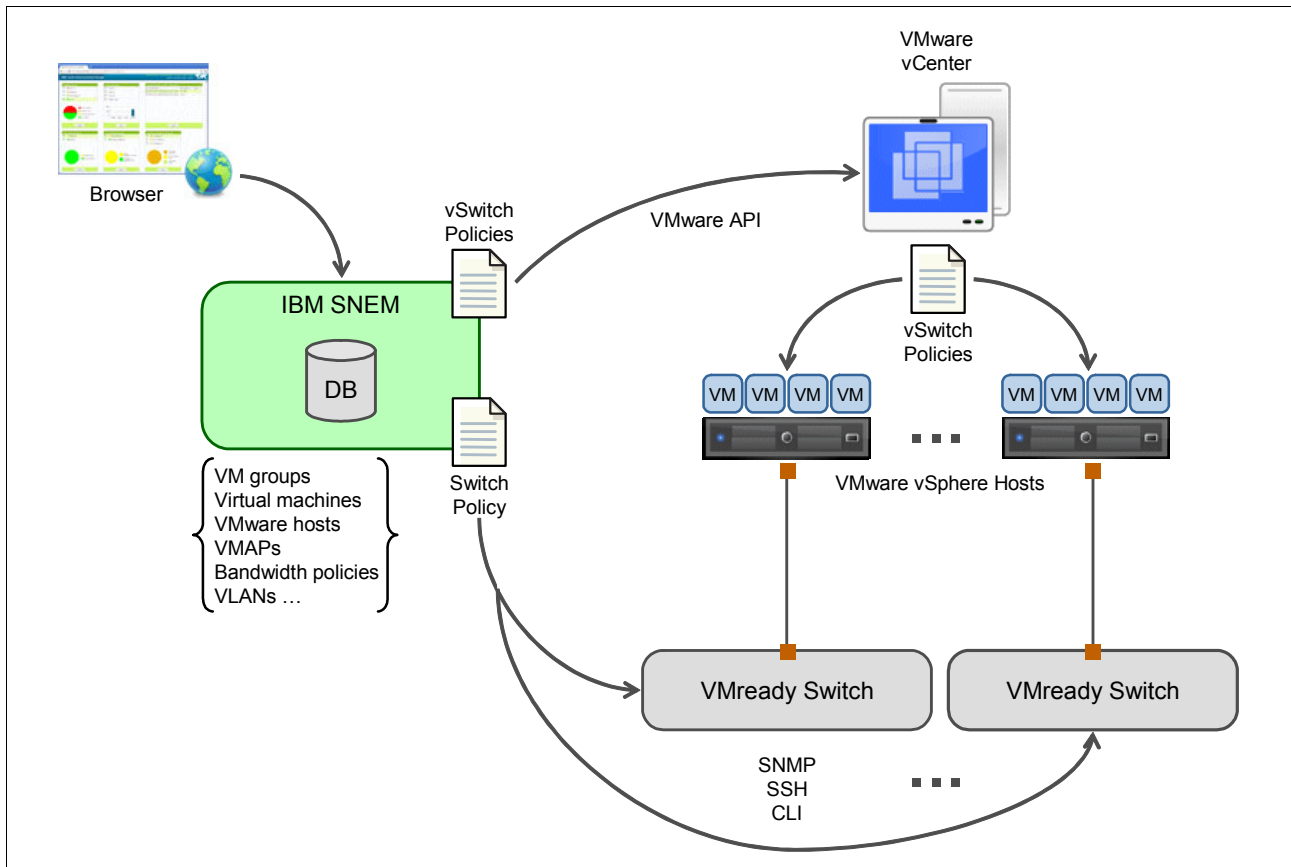


Figure 3-15 VMready with Virtual Vision

3.3 Monitoring and troubleshooting your VMready installation

IBM System Network Element Manager provides several useful tools for monitoring VMready. Tools are available both for VMready with Edge Virtual Bridging (IEEE 802.1Qbg) and when not using Edge Virtual Bridging.

3.3.1 Monitoring VMready with Edge Virtual Bridging (IEEE 802.1Qbg)

Click **Device Console** → **Monitor** → **EVB** to view several aspects of your Edge Virtual Bridging setup as shown in Figure 3-16.

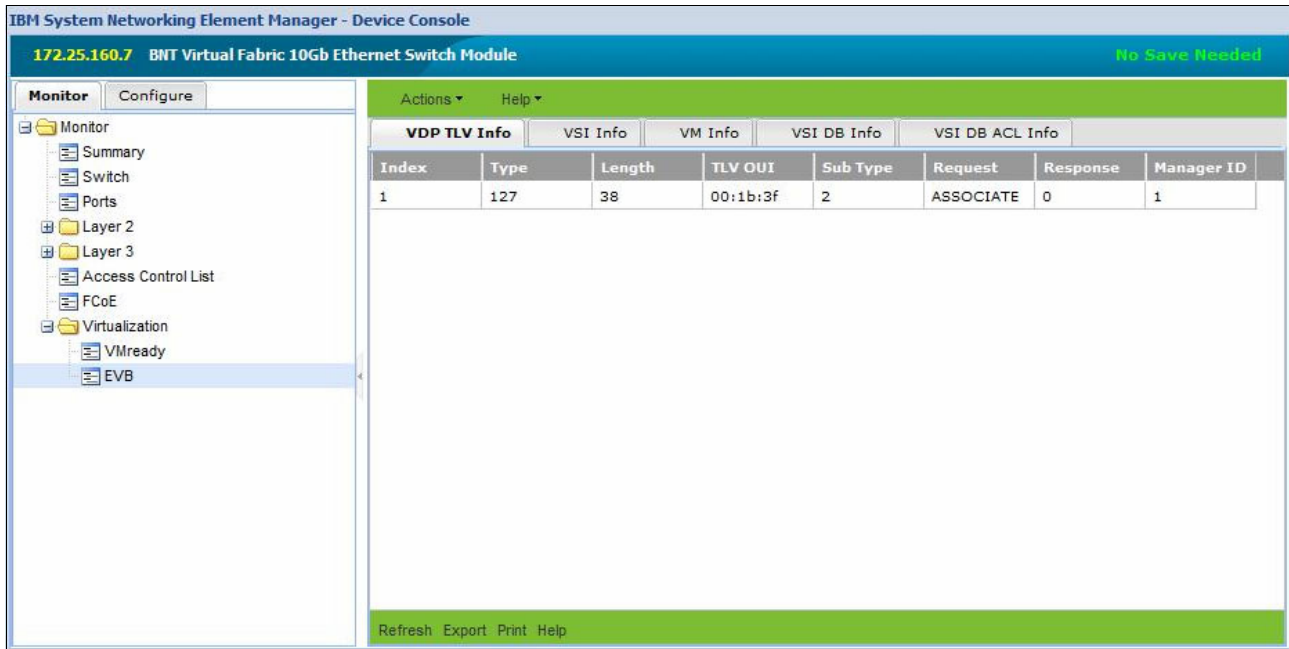


Figure 3-16 Monitoring VMready with Edge Virtual Bridging (IEEE 802.1Qbg)

3.3.2 Monitoring VMready when not using Edge Virtual Bridging

Click **Device Console** → **Monitor** → **Virtualization** → **VMready** to view several aspects of your VMready setup. Figure 3-17 shows the port information.

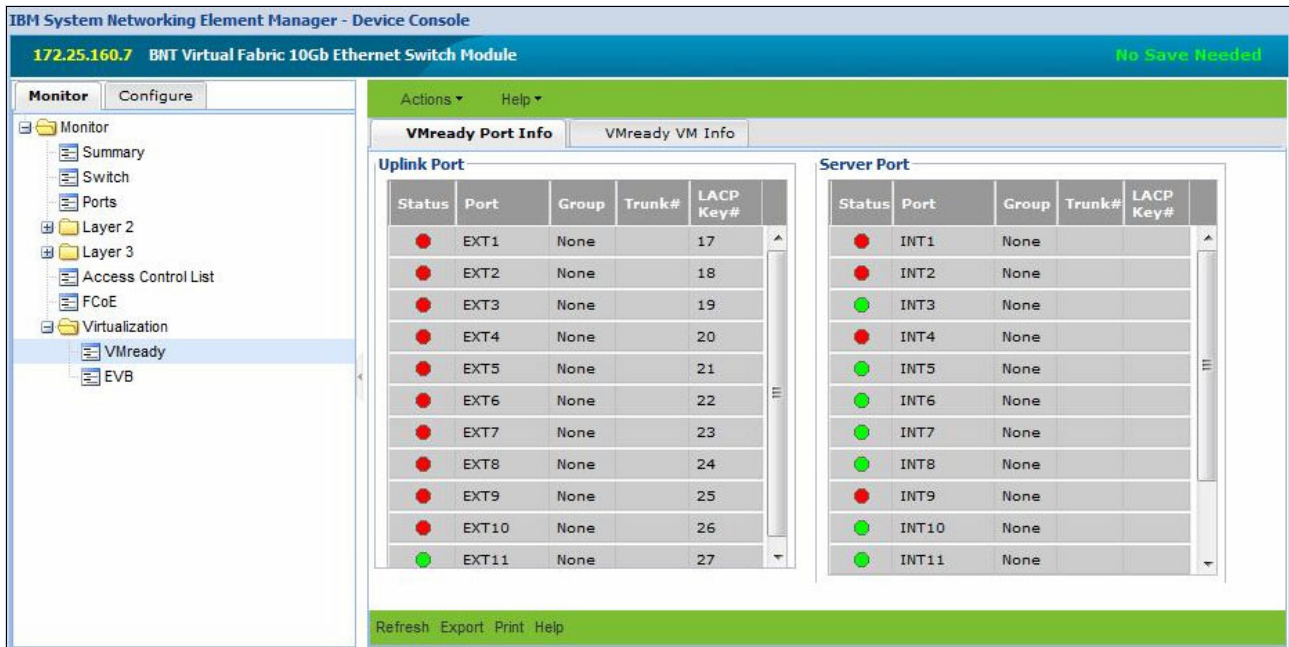


Figure 3-17 Device Manager Virtualization VMready port information

Figure 3-18 shows the VM information.

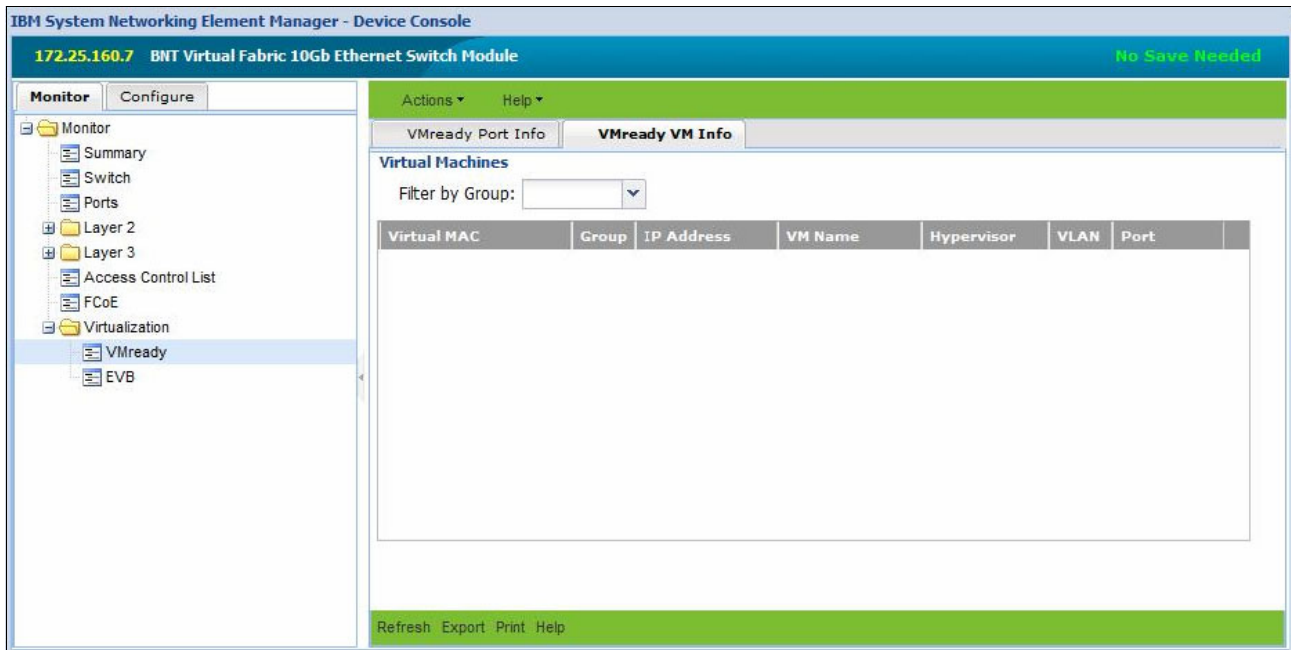


Figure 3-18 Device Manager Virtualization VMready VM information



Part 2

Implementing a VM-aware network

This part addresses how to implement a VM-aware network with these specific implementation scenarios:

- ▶ Implementing VMready to support VMware
- ▶ Implementing VMready to support PowerVM
- ▶ Implementing VMready to support KVM
- ▶ Implementing VMready to support Hyper-V and other virtualization environments



Implementing VMready to support VMware

This chapter describes the steps involved in implementing VMready for VMware environments. This chapter includes the following sections:

- ▶ Overview of VMready for VMware
- ▶ Implementation scenario
- ▶ Implementing VMready
- ▶ Implementing VMready with IEEE 802.1Qbg support

4.1 Overview of VMready for VMware

IBM VMready allows you to configure virtual machine networking at the physical switch layer. It provides a seamless interface for configuring both physical and virtual server networking. In a VMware environment, VMready ensures that VMs maintain network policies as they migrate by using a technology called NMotion. With this process, you no longer need to reconfigure individual switch ports for virtual machine networking. In addition, VMready configures the vSwitches within the hypervisor to match the physical switch configuration down to the VM level.

After configuration, the VMready switch ensures that connectivity and appropriate network policies are enforced for both virtual and physical servers.

4.1.1 VMware networking

VMware is a leader in the x86 hypervisor marketplace. They provide a rich set of capabilities within the vSphere product family for virtual machine deployment, configuration, and management. As virtual infrastructures have become increasingly demanding, VMware has met the new requirements with subsequent releases of the hypervisor. The latest releases of VMWare vSphere natively include in-host networking in the form of the standard virtual switch and a distributed virtual switch.

Traditionally with VMware vSphere environments, the network administrators have one of two levels of involvement. They could study VMware's implementation of virtual networking to understand the true capabilities of the bundled vSwitches. This implementation involved becoming skilled on a new set of interfaces. More commonly, they were limited to the configuration of the physical switches only. Therefore, they had to surrender configuration of the network access layer to the virtual server administrator.

In the latter scenario, the server administrator would need to learn the specifics of access layer networking within the confines of a VMware specific implementation. They would not only need to become familiar with access layer networking, but also how it translates into a vSphere virtual switch environment. Only then could they ensure that the appropriate networking parameters are applied at the network access layer. This process was similar to parameters that are implemented by the network administrator for traditional physical servers.

In both these scenarios, either the network administrator or server administrator (or both) had to compromise to configure virtual server infrastructures.

VMready in VMware vSphere environments allows enhanced visibility and control of virtual machine networking. It allows network administrators to regain control of the network access layer. These benefits are achieved while still retaining for the benefits of network switching as close to the virtual machine as possible.

Remember: At the time of writing, VMready supports automatic configuration of port groups on vSphere standard switches. VMready support for vSphere distributed switches is planned for a future firmware release.

4.1.2 VMready capabilities in a VMware environment

VMready provides enhanced granularity to physical switch network capabilities. It extends the ability of the switch from specifying network parameters at the physical port level down to the virtual machine level. The following functions are available when configuring parameters per virtual machine port:

- ▶ Virtual LAN (VLAN) membership
- ▶ Traffic shaping and monitoring
- ▶ Access Control Lists (ACLs)
- ▶ Quality of service (QoS) attributes

Remember: At the time of writing, VLAN membership and traffic shaping parameters can be enforced both at the physical switch and the vSphere standard switch level. ACLs and QoS attributes apply to virtual machine traffic only when it traverses a physical switch port.

4.1.3 VMready implementation overview

Configuring VMready in a VMware vSphere environment requires the tasks in Figure 4-1 be performed by the system and network administrators. Some tasks are required only for the initial setup of VMready, whereas others might be repeated for each VM deployment.

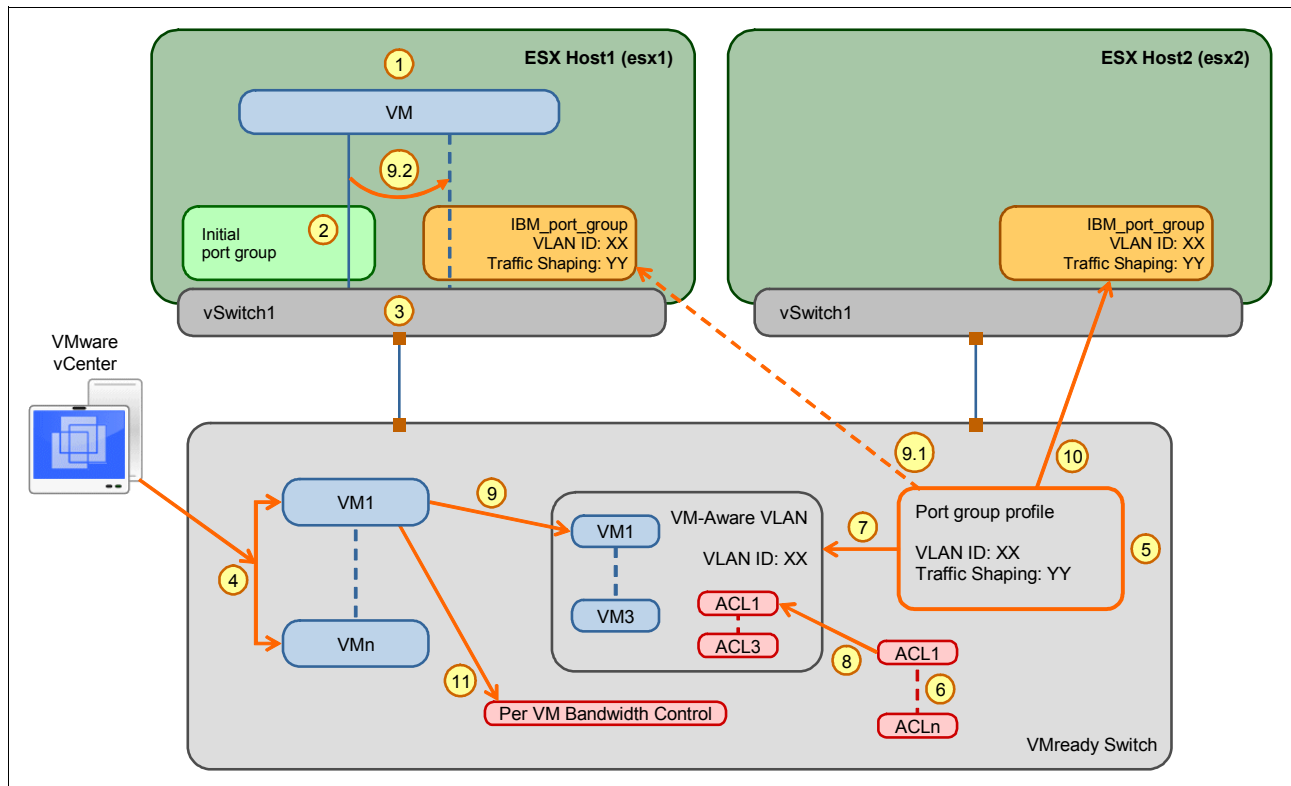


Figure 4-1 VMready conceptual configuration

Table 4-1 lists the implementation steps of VMready.

Table 4-1 Implementation steps

Server Administrator	<p>1. Creates the virtual machine.</p> <p>2. Assigns the VM to a default port group on a vSphere standard switch. This can be a staging / temporary port group that facilitates the server build process.</p> <p>3. Ensures that the port group to which the virtual machine is connected is on an appropriate vSwitch that has uplinks to the VMready switch.</p>
Network Administrator	<p>4. Enables VMready and configures VMWare vCenter credentials on the physical switch.</p> <p>This configuration allows the VMready switch to access information about the VMs and hosts in the vCenter environment. This initial configuration is required only once per VMready switch.</p> <p>5. Determines the network policies required for each of the new VMs and whether new port group profiles or policy definitions are required. If required, creates VLAN-centric port group profiles for each type of VM. At a minimum, the port group profile must contain a unique VLAN ID. Port-group level traffic shaping parameters can also be specified.</p> <p>A port group profile directly translates to a port group configuration on a vSphere standard switch.</p> <p>6. Creates any ACLs that apply to the groups of VMs.</p> <p>7. Associates each port group profile created in step 2 to a unique VM group container. This container takes on the identity of the VLAN specified in the port group profile and acts as a VM-Aware VLAN.</p> <p>Only one profile can apply per VM group container.</p> <p>8. Attaches the ACLs created in step 6 to the appropriate VM group container.</p> <p>Multiple ACLs can apply to a single VM Group.</p> <p>9. Adds individual VMs to their appropriate VM group based on the target VLAN. The switch is aware of all VMs in the VMware vCenter inventory. Therefore, VMs can be added to VM groups on any particular VMready switch even if they are currently on a host attached to a different physical switch.</p> <p>The VMready switch automatically creates the port group on the VMware virtual switch to which the VM will belong. This process uses API calls to the VMware vCenter server. These calls are based on the VLAN ID and traffic shaping configuration specified in the port group profile associated with the VM group. Automatic creation of a port group occurs on the ESX host on which the VM currently is located.</p> <p>The VMready switch also updates the port group assignment of the VM's virtual network adapter to the newly created port group.</p> <p>10. Exports the port group profile to any other ESX host in the cluster where the VM can migrate to. This process ensures that the appropriate port groups exist on all hosts to which the VM can migrate.</p> <p>11. Applies any per-VM traffic shaping policy that was not applicable at the port group level.</p>

4.2 Implementation scenario

This section uses a scenario-based approach to illustrate the actual steps required to enable VMready in a VMware environment. The scenario involves a corporate IT environment.

4.2.1 Example IT policies

The following are examples of IT policies that can exist in a corporate environment where departments use shared IT resources. These policies are often managed by a centralized IT infrastructure team.

Policy 1

Strict layer 2 network separation must exist between applications that belong to different business units.

Policy 2

No web application servers should be accessible on standard http ports.

Policy 3

Business units that host applications on shared IT infrastructure are restricted in the amount of the shared resource that is made available to them. The intention of the policy is two fold:

- ▶ To allow fair sharing of available resources in multi-tenant environments to prevent one department from monopolizing IT resources to the detriment of another.
- ▶ To ensure that departments are allocated resources in proportion to their contribution to IT budgets.

4.2.2 Environment overview

In the example, a simple set of 3 VMs is used to illustrate the application of the IT policies described in 4.2.1, “Example IT policies” on page 49. The hypothetical organization has two business units with requirements to host applications on the core IT infrastructure. These units are the following departments:

- ▶ Finance requires two servers to host a two tier web application on servers named *WEB* and *DB*.
- ▶ HR requires a single server to host their infrastructure file server role. This VM is named *FILE*.

4.2.3 Initial configuration

As shown in Figure 4-2 the implementation scenario uses two host systems, three VMs (FILE, WEB, and DB), and a client system. The client system is used to test the impact of implementing network policies and determining whether those policies move with VMs as they are migrated.

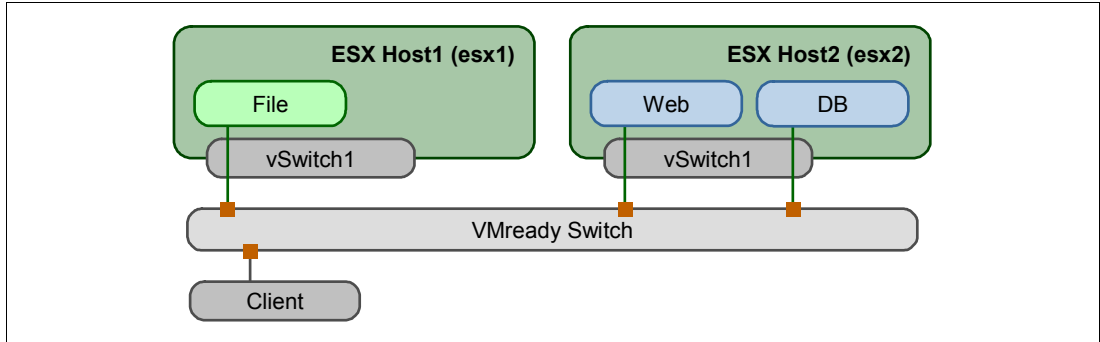


Figure 4-2 Implementation scenario

Figure 4-3 shows the physical layout of the environment before VMready is implemented.

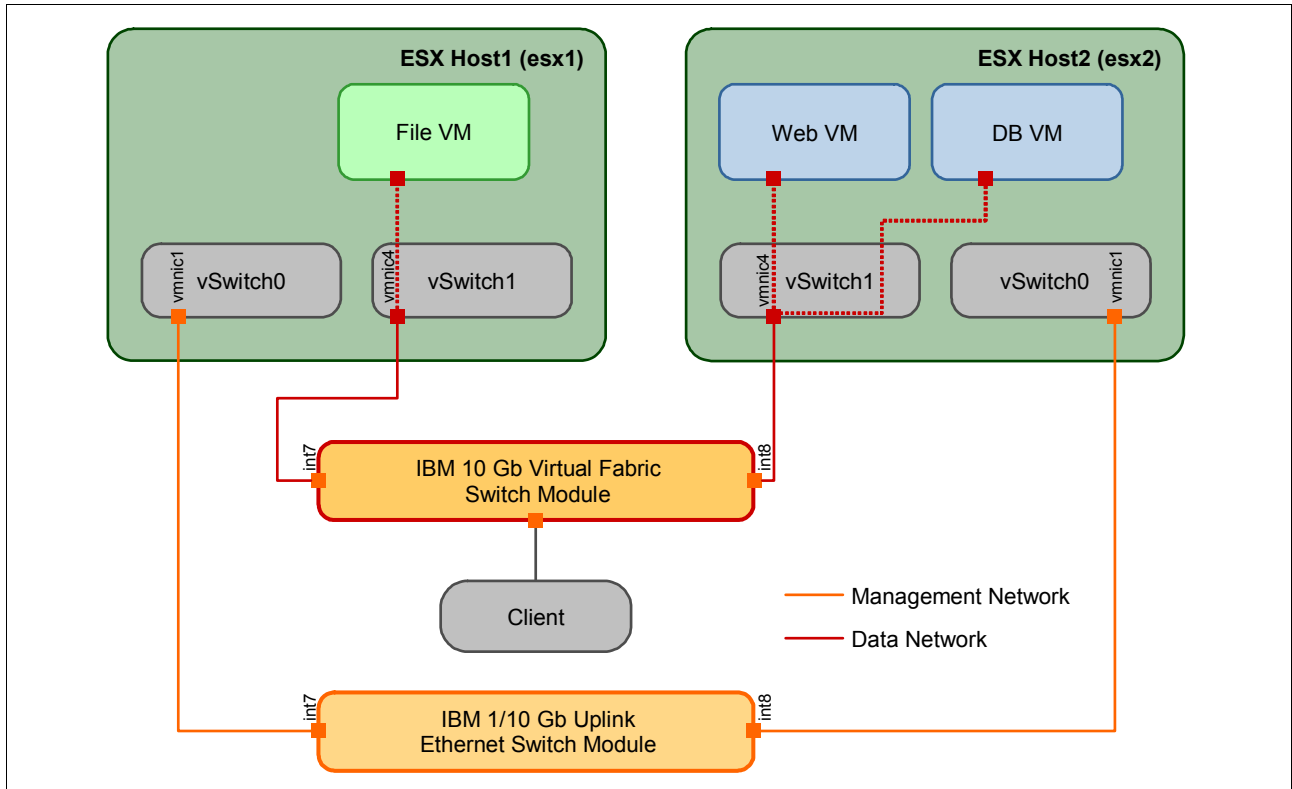


Figure 4-3 Physical network connectivity

Each host has a 1 Gbps connection to an IBM BNT 1/10 Uplink Switch Module and a 10 Gbps connection to an IBM BNT 10 Gb VF Switch Module. The 1 Gbps connection is used solely for ESX service console traffic. It therefore does not need to be configured for VMready. VMready is configured on the IBM BNT 10 Gb VF Switch Module because all VM traffic will traverse this switch.

The first 10 Gb interface of esx1 is connected to physical switch port INT7. The first 10 gigabit interface of esx2 is connected to physical switch port INT8. The client system used for testing the configuration is connected to port INT6 of the IBM BNT 10 Gb VF Switch Module.

The environment is configured to use VMware vSphere 5.0 and is managed by an external vCenter 5.0 server. All VMs are initially attached to the default port group VM Network on vSwitch1.

The firmware and software levels used are shown in Table 4-2.

Table 4-2 Software versions used for VMReady

Component	Software
IBM BNT 10 Gb Virtual Fabric Switch Module	IBM Networking OS 6.8.0.66
ESX Host	ESXi 5.0 build 441354
vCenter Server	vCenter 5.0.0 build 455964

Figure 4-4 shows the initial view from vCenter of esx1.

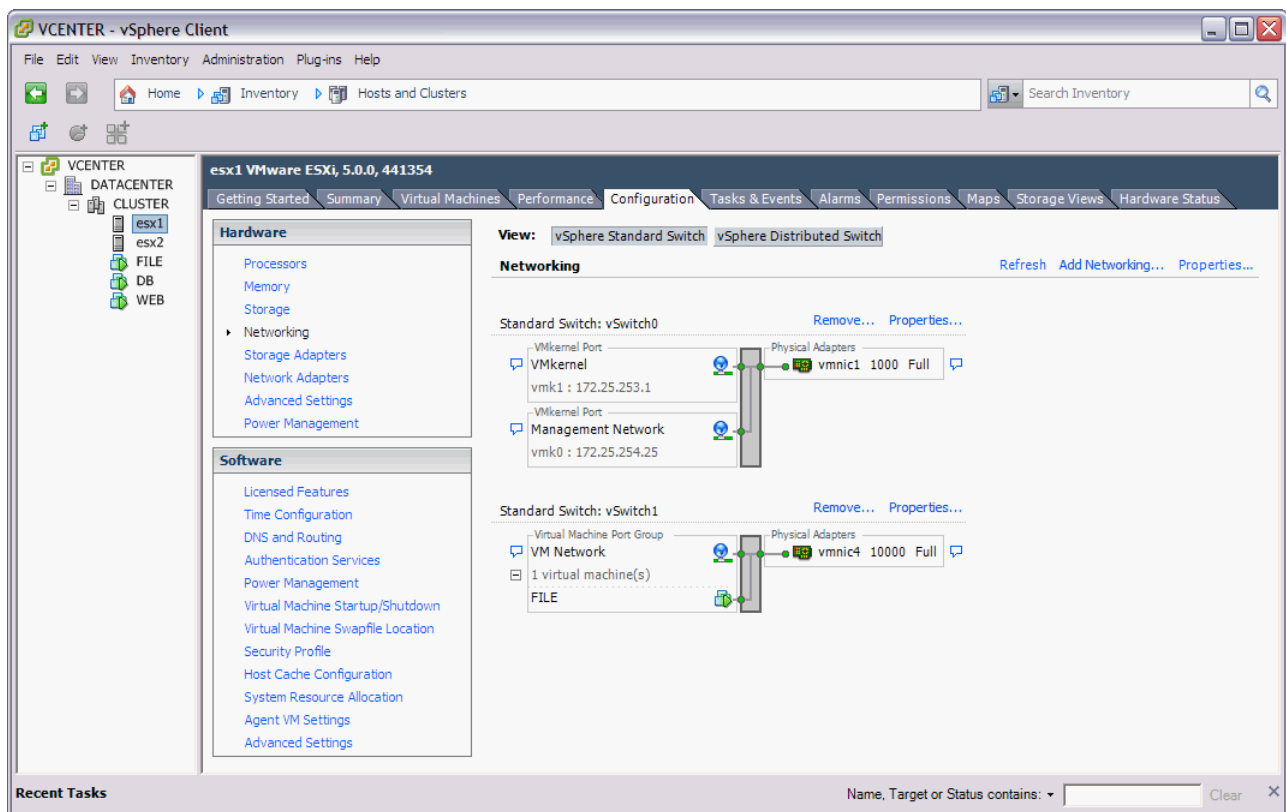


Figure 4-4 Initial network configuration of esx1

Figure 4-5 shows the initial view from vCenter of esx2.

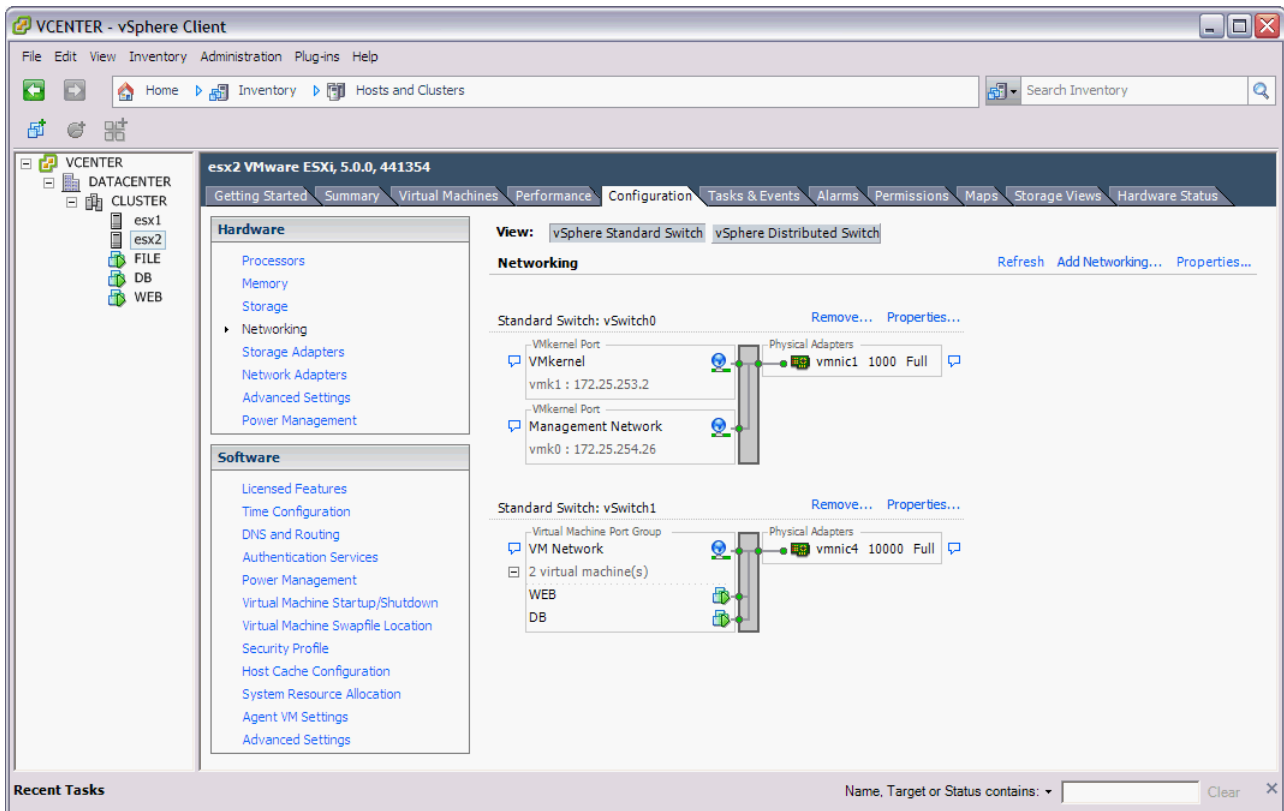


Figure 4-5 Initial network configuration of esx2

At this stage, the configuration of the servers violates the corporate IT policies defined in 4.2.1, “Example IT policies” on page 49:

1. No layer 2 separation exists between the Finance and HR VMs. In Figure 4-6, WEB and FILE servers are able to exchange traffic due to lack of VLAN segregation.

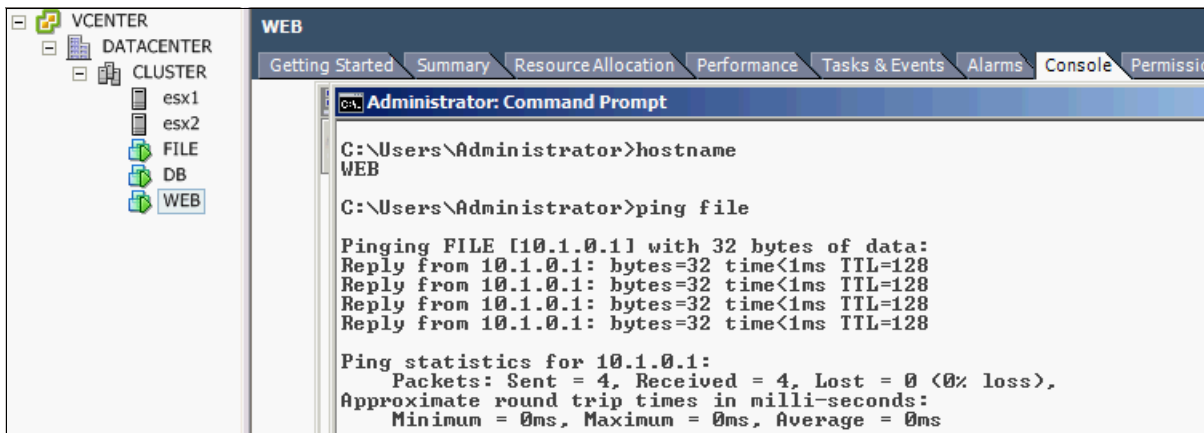


Figure 4-6 No segregation of HR and finance servers

2. An external client connected to the physical switch can connect to the WEB server on the default http port: 80 as shown in Figure 4-7.

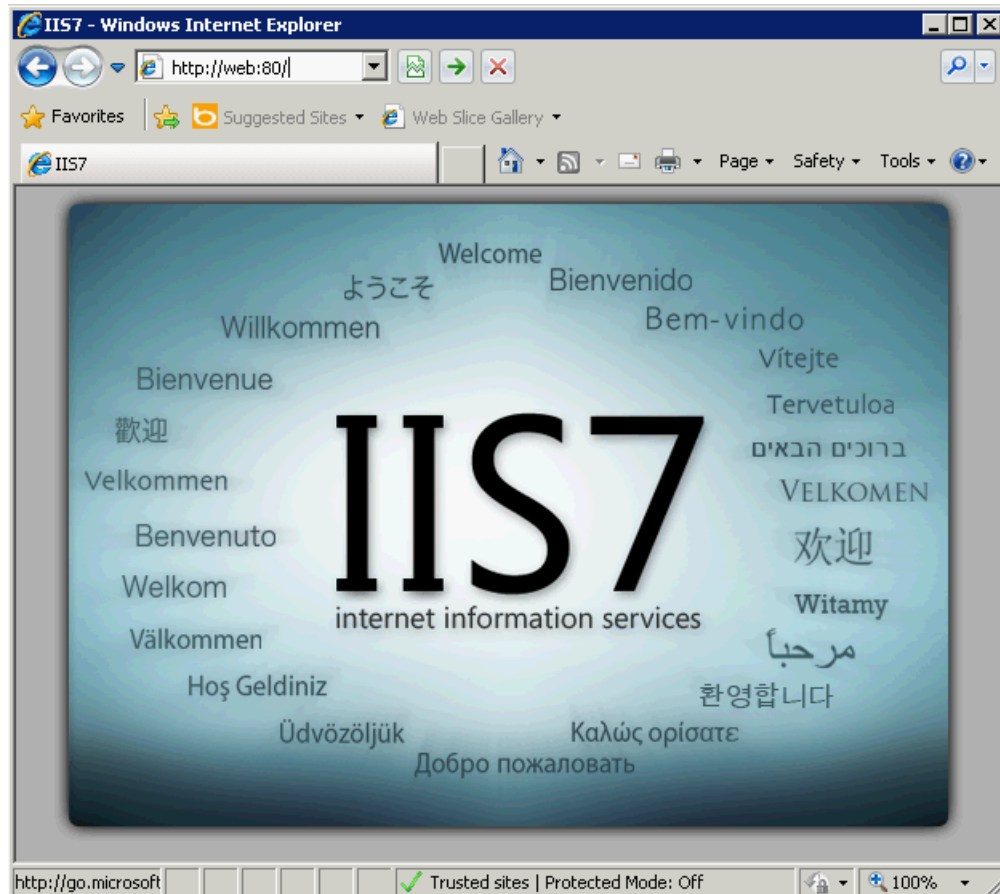


Figure 4-7 WEB server successful in serving web pages on standard HTTP port (80)

- The HR department file server is unrestricted in its use of available network bandwidth as shown in Figure 4-8.

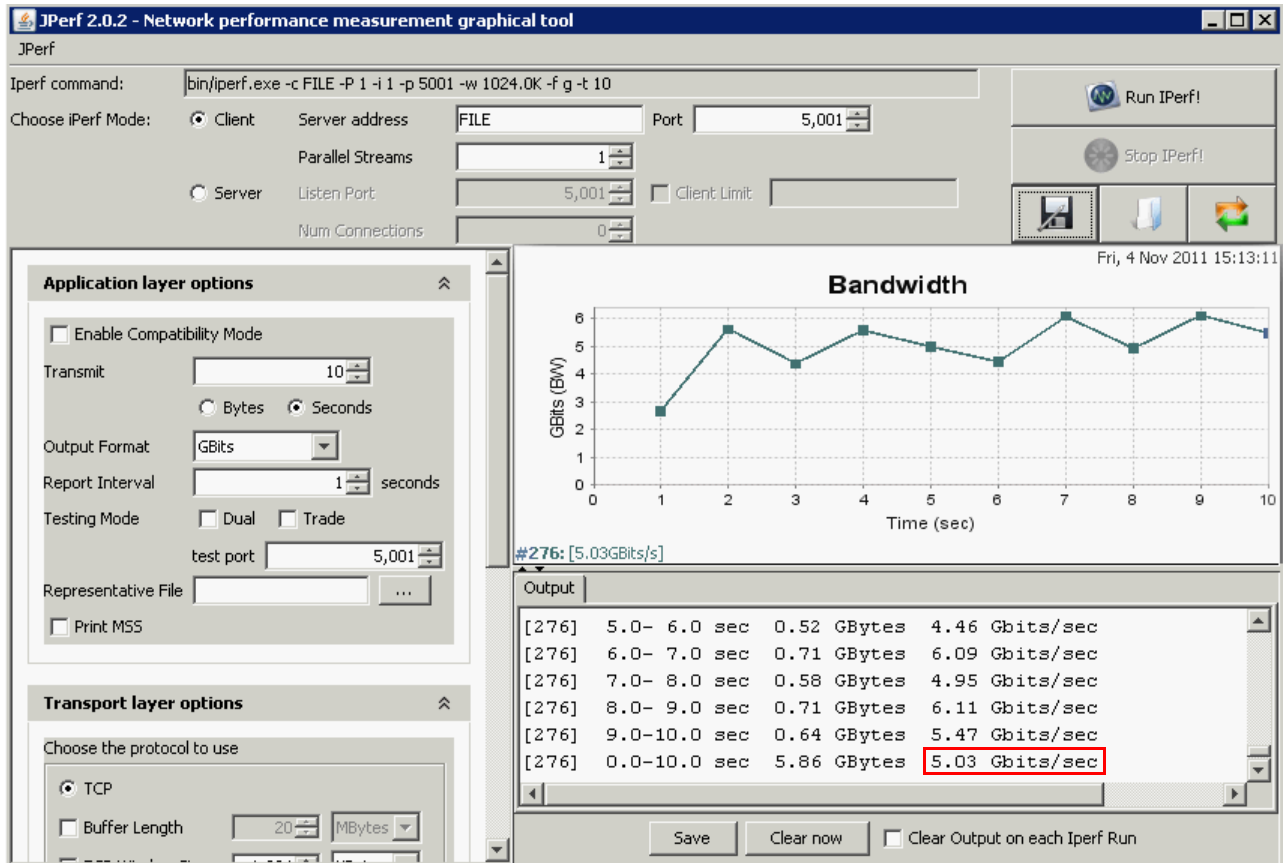


Figure 4-8 JPerf output from external client that is connected to FILE

Tip: These IT policies might have implications on server and storage configurations as well. However, this book focuses on their application within the context of datacenter networking.

4.2.4 Target scenario

To meet the IT policies, you must implement these configuration changes.

- ▶ Connect the finance WEB and DB servers to the finance web application VLAN: 1100.
- ▶ Connect the HR FILE server to the HR infrastructure VLAN: 1200.
- ▶ Use a group ACL on the Finance web application VLAN to prevent all incoming traffic destined for port 80.
- ▶ Apply a per-VM bandwidth policy to the FILE server to restrict the amount of bandwidth it can consume to 1 Mbps.

Figure 4-9 shows a target configuration that meets the minimum requirements of the organization's IT policies.

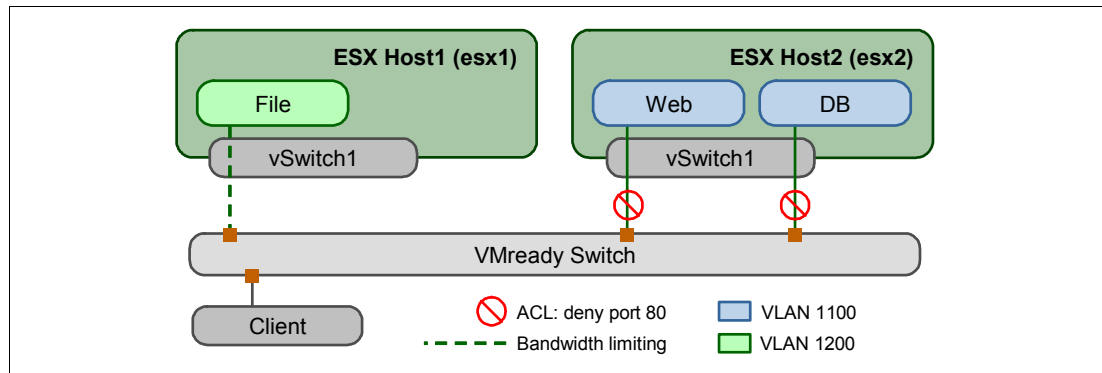


Figure 4-9 Target VMready configuration

4.3 Implementing VMready

This section defines specific objects that need to be configured to enable VMready functionality within the physical switch and the workflow.

4.3.1 Definitions

vmprofile	This contains the vSphere standard port group configuration. It includes a mandatory VLAN ID and an optional traffic shaping policy. It corresponds to the port group configuration that is pushed by using the vCenter API.
vmap	This is a specialized ACL that can apply to VLANs and vmgroups.
vmgroup	This is a VM-Aware enhanced VLAN in which we can place VMs and associate vmaps. A VLAN ID can be directly assigned to a vmgroup. However, if automatic vSwitch configuration is wanted, a vmprofile must be associated with a vmgroup instead. The vmgroup then occupies the VLAN ID defined in the vmprofile. Only one vmprofile can apply to each vmgroup. After a vmgroup is associated with a VLAN ID (either directly or through a vmprofile), it becomes the new point of configuration for that VLAN. Physical switch ports, static trunks, and LACP trunks can also be added to a vmgroup just like a regular VLAN.

4.3.2 Configuration workflow

In light of these terms, the workflow original diagram from Figure 4-1 on page 47 can be modified as per Figure 4-10.

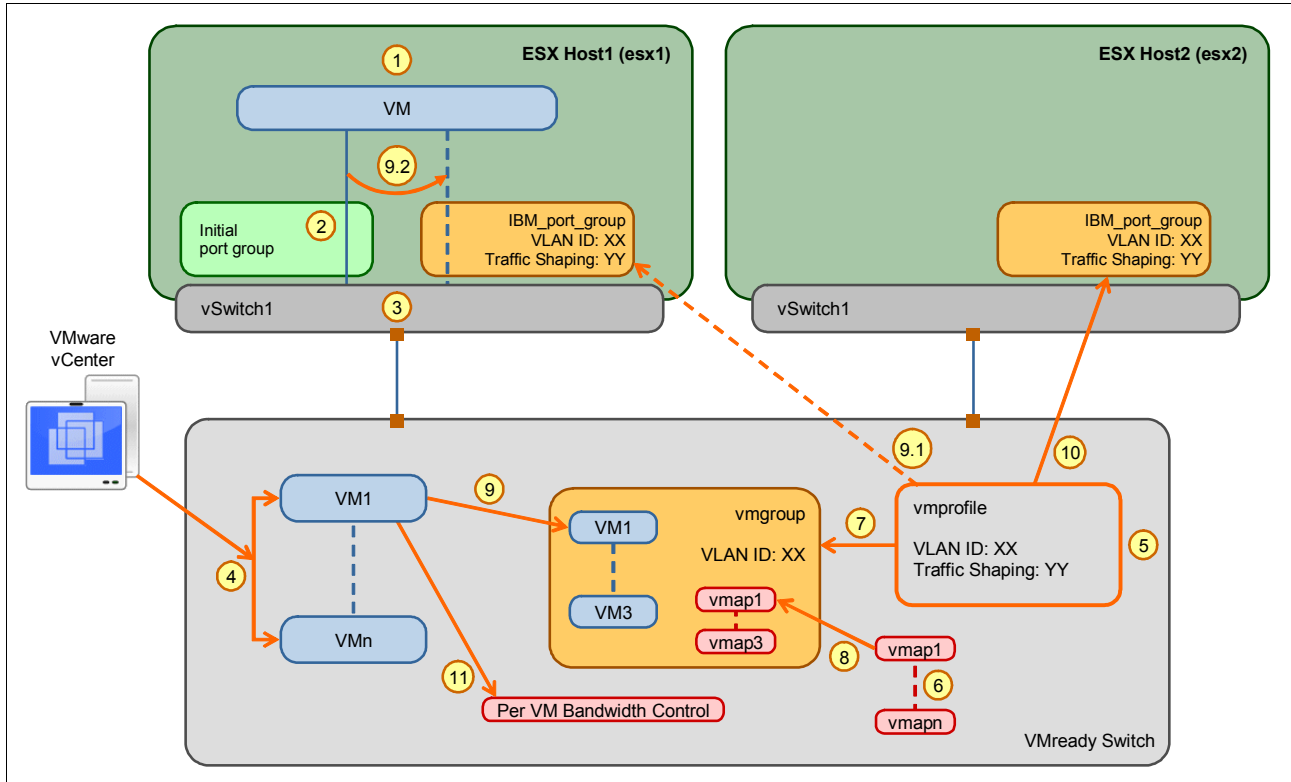


Figure 4-10 VMready implementation steps

4.3.3 Entering configuration mode

To configure the VMready switch, enter configuration mode, then open a command-line interface (CLI) session to the IBM BNT 10 Gb VF switch by using the industry standard CLI. Thereafter, enter configuration mode as shown in Example 4-1.

Attention: CLI commands might vary slightly in subsequent firmware releases due to updates in standards terminology and usability improvements.

Example 4-1 Entering configuration mode

```
Router>enable
```

```
Enable privilege granted.
```

```
Router#configure terminal
```

```
Enter configuration commands, one per line. End with Ctrl/Z.
```

```
Router(config)#
```

The following configuration steps must be performed by the network administrator. The assumption is that the server administrator has already completed steps 1, 2, and 3.

4.3.4 Enabling VMready

Use this global switch command to enable VMready

```
virt enable
```

Thereafter, vCenter configuration can be entered by using this command:

```
virt vmware vcspec A.B.C.D username
```

A prompt for the password of the vCenter administrative account follows the command.

After the command is entered, the switch automatically logs on to vCenter and obtains information about virtual machines, hosts, and virtual switches (Example 4-2).

Example 4-2 Enabling VMready

```
Router(config)#virt enable
Router(config)#virt vmware vcspec 172.25.254.22 Administrator noauth
Enter Virtual Center password:
```

Tip: VLAN tagging is enabled by default on all internal ports on the IBM BNT 10 Gb VF switch module. However, this is not usually the case for RackSwitch products. VLAN tagging needs to be enabled for VMready to function on a port.

After vCenter credentials are configured, query for vCenter related information directly from the switch by using the following commands. The **show virt vmare** command is shown in Example 4-3.

Example 4-3 Showing VMware settings

```
Router(config)#show virt vmware

Current VMware-specific settings
-----

ESX/ESXi-to-vCenter heartbeat UDP port number: 902

Virtual Center 172.25.254.22, User name "Administrator", No Authentication
```

The **show virt vmare vms** command is shown in Example 4-4.

Example 4-4 Showing VMware vms

```
Router(config)#show virt vmware vms
UUID                               Name(s), IP Address
-----
42199620-df9d-abaa-9dc9-4a4be17e28af  FILE, 10.1.0.1
421996de-70aa-cd72-599d-0a413773b8e1  DB, 10.0.0.2
4217c5ec-a748-aaa6-0c23-498c950f9347  WEB, 10.0.0.1
```

The **show virt vmare hosts** command is shown in Example 4-5.

Example 4-5 Showing VMware hosts

```
Router(config)#show virt vmware hosts
UUID                               Name(s), IP Address
-----
```

```
65415825-8a16-3ca1-bc72-46a5d7200369 esx1
c0a52aca-4066-b601-d7e0-001a64cf36fe esx2
```

The **show virt vmare showhost esx1** command is shown in Example 4-6

Example 4-6 Showing VMware host details

```
Router(config)#show virt vmare showhost esx1
```

Vswitches available on the host:

```
vSwitch0
vSwitch1
```

Port Groups and their Vswitches on the host:

```
VMkernel                vSwitch0
Management Network      vSwitch0
VM Network                vSwitch1
```

Detailed information about host and VM interfaces on this hypervisor:

```
-----
MAC Address      00:1a:64:5b:16:54
Port             N/A
Type             VM Kernel
IP Address       172.25.254.25
Host Name        esx1
Host UUID        65415825-8a16-3ca1-bc72-46a5d7200369
Vswitch          vSwitch0
Port Group       Management Network
VLAN ID          0
-----
```

```
-----
MAC Address      00:50:56:78:90:2b
Port             N/A
Type             VM Kernel
IP Address       172.25.253.1
Host Name        esx1
Host UUID        65415825-8a16-3ca1-bc72-46a5d7200369
Vswitch          vSwitch0
Port Group       VMkernel
VLAN ID          0
-----
```

```
-----
MAC Address      00:50:56:99:25:73
Port             INT7
Type             Virtual Machine
VM vCenter Name  FILE
VM OS hostname  FILE
VM IP Address    10.1.0.1
VM UUID          42199620-df9d-abaa-9dc9-4a4be17e28af
Current VM Host  esx1
Vswitch          vSwitch1
Port Group       VM Network
VLAN ID          0
-----
```

4.3.5 Defining vmpfiles

Create two vmpfiles to provide port group definitions for the Finance and HR VMs as shown in Example 4-7:

- ▶ A profile named FIN-WEBAPP01 configured with VLAN 1100. This profile applies to the vmgroup that contains the Finance department web application servers: WEB and DB.
- ▶ A profile named HR-INFRA01 configured with VLAN 1200 for the VM hosting the HR department file server.

Example 4-7 Defining vmpfiles

```
Router(config)#virt vmpfile FIN-WEBAPP01
Router(config)#virt vmpfile edit FIN-WEBAPP01 vlan 1100
Router(config)#virt vmpfile HR-INFRA01
Router(config)#virt vmpfile edit HR-INFRA01 vlan 1200
```

You can define a traffic shaping policy within the **vmpfile** itself using the **virt vmpfile edit HR-INFRA01 shaping** command. This command applies traffic shaping at the vswitch port group level for outbound traffic.

Alternatively, the VMready physical switch can apply VM policy bandwidth control on a per VM basis for both egress and ingress traffic. This is done in the final step.

Remember: Although vmpfiles are used in this example, it is not compulsory to define vmpfiles for VMready to work. VLAN IDs can be directly configured on vmgroups instead of vmpfiles. Direct configuration is useful in environments where standard vSwitch port groups are already established and you do not want to configure them with the physical switch infrastructure.

4.3.6 Defining group ACLs (vmaps)

Define a single group ACL (vmap) using an available vmap number: 11 as shown in Example 4-8. Define it to deny all traffic destined for port 80.

Example 4-8 Denying port 80 ACL

```
Router(config)#access-control vmap 11 tcp-udp destination-port 80
Router(config)#access-control vmap 11 action deny
```

4.3.7 Applying vmpfiles to vmgroups

Associate each **vmpfile** to a unique **vmgroup** as shown in Example 4-9. The **vmgroup** takes over the identity of the VLAN ID in the associated vmpfile to create a VM-Aware VLAN.

Example 4-9 Associating vmpfiles with vmgroups

```
Router(config)#virt vmgroup 11 profile FIN-WEBAPP01
```

VLAN 1100 was assigned to STG 76.

```
Router(config)#virt vmgroup 21 profile HR-INFRA01
```

VLAN 1200 was assigned to STG 48.

The action of associating a vmaprofile with a vmgroup activates the VLAN configured in the vmaprofile. Per VLAN spanning tree is enabled by default on the switch, so the switch assigns VLANs 1100 and 1200 to spanning tree groups.

4.3.8 Applying vmaps to vmgroups

Allocate the vmap defined earlier to the group that contains the finance web application servers, WEB and DB, as shown in Example 4-10.

Example 4-10 Assigning vmap to vmgroup

```
Router(config)#virt vmgroup 11 vmap 11
```

4.3.9 Adding VMs to vmgroups

Add VMs to your vmgroups as shown in Example 4-11. After a VM is added to a vm group, port groups are created automatically on the host where the VM is running. VMs can be added based on the VM name, IP address, MAC address, or UUID. When a VM has multiple virtual network adapters, specify the exact target MAC address. In the example, all VMs have a single network adapter, so use the VM's name.

Example 4-11 Adding VMs to vmgroups

```
Router(config)#virt vmgroup 11 vm WEB
Rescanning data center for MAC address 00:50:56:97:5c:81. Please wait.
Router(config)#virt vmgroup 11 vm DB
Router(config)#virt vmgroup 21 vm FILE
```

4.3.10 Exporting vmaprofiles to ESX hosts

Export vmaprofiles to all other esx hosts in the cluster. To query esx host configuration, use the commands shown in Example 4-12.

Example 4-12 View VMware hosts

```
Router#show virt vmware hosts
UUID                               Name(s), IP Address
-----
65415825-8a16-3ca1-bc72-46a5d7200369 esx1
c0a52aca-4066-b601-d7e0-001a64cf36fe esx2
```

```
Router#show virt vmware showhost esx1
```

Vswitches available on the host:

```
vSwitch0
vSwitch1
```

Port Groups and their Vswitches on the host:

```
VMkernel           vSwitch0
Management Network vSwitch0
VM Network          vSwitch1
IBM_HR-INFRA01     vSwitch1
```

Detailed information about host and VM interfaces on this hypervisor:

MAC Address	00:1a:64:5b:16:54
Port	N/A
Type	VM Kernel
IP Address	172.25.254.25
Host Name	esx1
Host UUID	65415825-8a16-3ca1-bc72-46a5d7200369
Vswitch	vSwitch0
Port Group	Management Network
VLAN ID	0

MAC Address	00:50:56:78:90:2b
Port	N/A
Type	VM Kernel
IP Address	172.25.253.1
Host Name	esx1
Host UUID	65415825-8a16-3ca1-bc72-46a5d7200369
Vswitch	vSwitch0
Port Group	VMkernel
VLAN ID	0

MAC Address	00:50:56:99:25:73
Port	INT7
Type	Virtual Machine
VM vCenter Name	FILE
VM OS hostname	FILE
VM IP Address	10.1.0.1
VM UUID	42199620-df9d-abaa-9dc9-4a4be17e28af
Current VM Host	esx1
Vswitch	vSwitch1
Port Group	IBM_HR-INFRA01
VLAN ID	1200

The vmprofile “HR-INFRA01” has already been configured on host esx1, where the FILE server is located.

To export a vmprofile to an esx host, you need to exit the configuration terminal mode. Enter **exit** to return to the standard terminal.

Export the FIN-WEBAPP01 profile to server esx1, and the HR-INFRA01 profile to esx2 as shown in Example 4-13.

Example 4-13 Exporting vmprofiles to esx hosts

```
Router#virt vmware export FIN-WEBAPP01 esx1 vSwitch1
Successfully exported profile FIN-WEBAPP01 to host esx1.
Router#virt vmware export HR-INFRA01 esx2 vSwitch1
Successfully exported profile HR-INFRA01 to host esx2.
```

Consideration: The configuration information in a vmprofile directly corresponds to the configuration of a port group on a vSphere standard switch. At the time of writing, support for the VMWare distributed switch is planned for a future firmware release.

4.3.11 Assigning per-VM bandwidth policy

As the final configuration step, assign a per-VM bandwidth shaping policy to ensure that the HR FILE server does not use all bandwidth on a host. Apply it for traffic coming from the VM (txrate) and going to the VM (rxrate). Set the bandwidth limits to 1 Mbps.

Implement the policy per VM bandwidth control as shown in Example 4-14.

Example 4-14 Assigning the per-VM policy

```
Router(config)#virt vmpolicy vmbwidth FILE rxrate 1024 1024
Router(config)#virt vmpolicy vmbwidth FILE txrate 1024 1024
Router(config)#virt vmpolicy vmbwidth FILE bwctrl
```

4.3.12 Post implementation review

Confirm that the configuration was passed on to vCenter. Run commands from the network switch to view the new configuration. The `show virt vmware showhost hostname` command provides a list of vSwitches and port groups on each host and an overview of each virtual machine (Example 4-15). For more information, see 4.3.13, “NMotion” on page 66.

Example 4-15 Showing detailed VMware host information

```
Router(config)#show virt vmware showhost esx2
```

Vswitches available on the host:

```
vSwitch0
vSwitch1
```

Port Groups and their Vswitches on the host:

```
VMkernel                vSwitch0
Management Network     vSwitch0
VM Network               vSwitch1
IBM_FIN-WEBAPP01        vSwitch1
IBM_HR-INFRA01          vSwitch1
```

Detailed information about host and VM interfaces on this hypervisor:

```
MAC Address      00:1a:64:cf:36:fe
Port             N/A
Type             VM Kernel
IP Address       172.25.254.26
Host Name        esx2
Host UUID        c0a52aca-4066-b601-d7e0-001a64cf36fe
Vswitch          vSwitch0
Port Group       Management Network
VLAN ID          0
```

```
MAC Address      00:50:56:77:67:c8
Port             N/A
Type             VM Kernel
IP Address       172.25.253.2
```

```

Host Name          esx2
Host UUID          c0a52aca-4066-b601-d7e0-001a64cf36fe
Vswitch            vSwitch0
Port Group         VMkernel
VLAN ID            0

```

```

-----
MAC Address        00:50:56:97:5c:81
Port               INT8
Type               Virtual Machine
VM vCenter Name    WEB
VM OS hostname     WEB
VM IP Address      10.0.0.1
VM UUID            4217c5ec-a748-aaa6-0c23-498c950f9347
Current VM Host    esx2
Vswitch            vSwitch1
Port Group         IBM_FIN-WEBAPP01
VLAN ID            1100

```

```

-----
MAC Address        00:50:56:99:56:ec
Port               INT8
Type               Virtual Machine
VM vCenter Name    DB
VM OS hostname     DB
VM IP Address      10.0.0.2
VM UUID            421996de-70aa-cd72-599d-0a413773b8e1
Current VM Host    esx2
Vswitch            vSwitch1
Port Group         IBM_FIN-WEBAPP01
VLAN ID            1100

```

There are two new port groups on vSwitch1 on host esx2: IBM_FIN-WEBAPP01 and BNT_HR-INFRA01.

The tasks run by the VMready switch using the vCenter APIs can be viewed in the Tasks & Events tab in vCenter at the cluster or datacenter level (Figure 4-11).

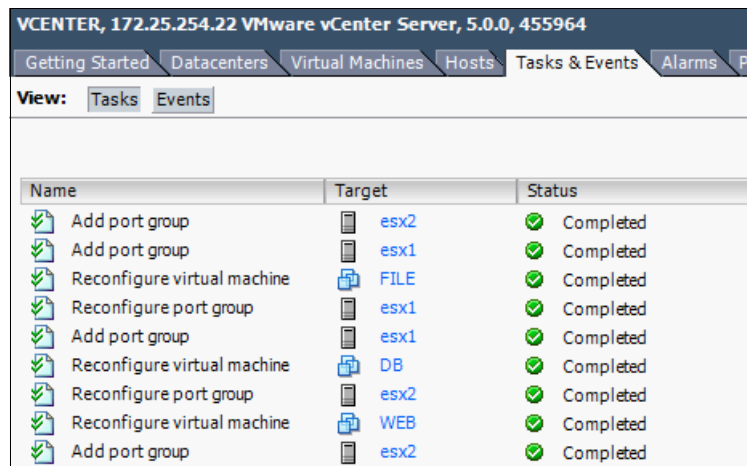


Figure 4-11 vCenter Tasks run

As ESX host, vSwitch, port group, and VM configurations are updated to reflect the network policies on the physical switch, the view from VMWare evolves. Figure 4-12 shows the new port groups created by the VMready switch on esx1, and the VMs that are configured to connect to those port groups.

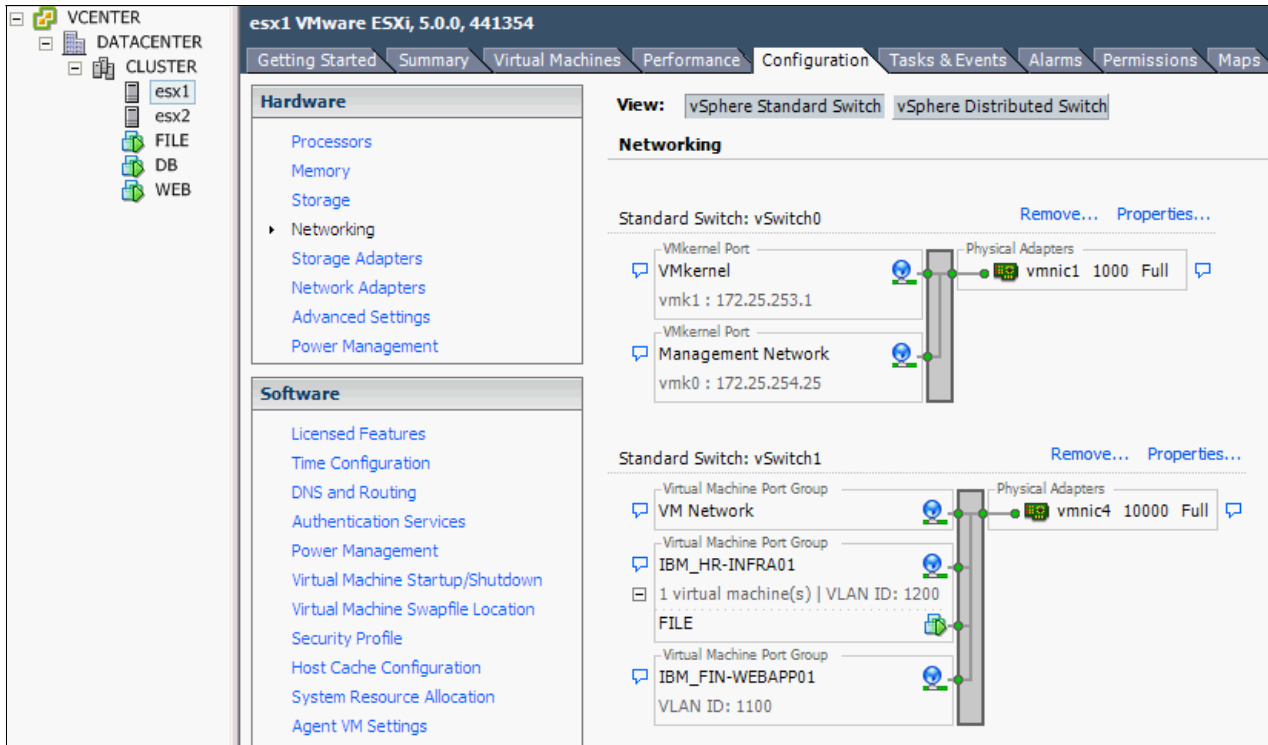


Figure 4-12 esx1 new network configuration

Figure 4-13 shows the new port groups created by the VMready switch on esx2.

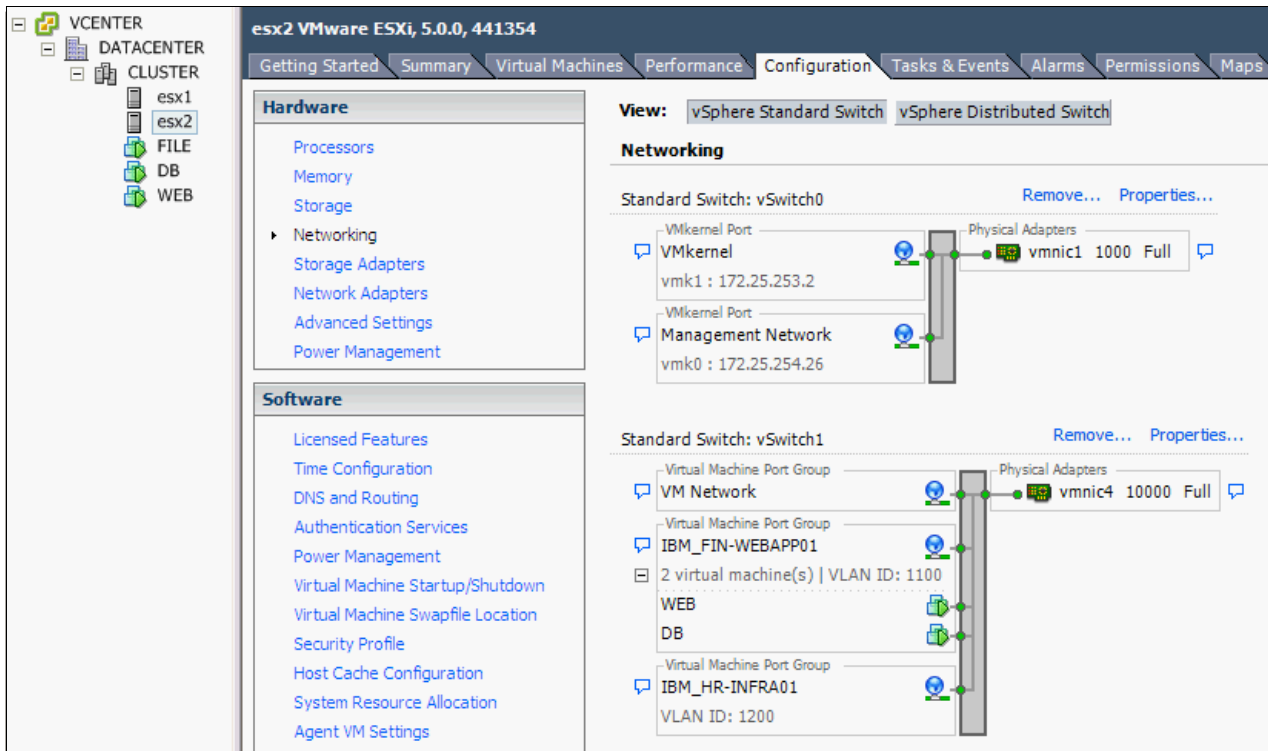


Figure 4-13 esx2 new network configuration

Recall that you associated the VMs WEB and DB to the vmprofile FIN-WEBAPP01. And the VM FILE was associated with vmprofile HR-INFRA01 in the physical switch. The corresponding changes are also propagated to vCenter to enforce a uniform network policy across virtual and physical switches.

A number of commands are available to help understand the new VM-aware network. Of particular interest are the **show virt vm** and **show interface information** commands. The **show virt vm** command shows a table of all active virtual machines as seen by the switch (Example 4-16).

Example 4-16 Showing virt vm output

```
Router>show virt vm
```

IP Address	VMAC Address	Index	Port	VM Group (Profile)
10.0.0.3	00:0c:29:06:54:75	2	IT 11	
10.1.0.2	00:0c:29:06:54:7f	1	IT 11	
10.0.0.1	00:50:56:97:5c:81	4	INT8	11 (FIN-WEBAPP01)
10.1.0.1	00:50:56:99:25:73	0	INT7	21 (HR-INFRA01)
10.0.0.2	00:50:56:99:56:ec	3	INT8	11 (FIN-WEBAPP01)

Number of entries: 5

This command also displays the vmgroup number, vmprofile name, and the current physical switch port that the virtual machine is visible on. The two Finance department VMs belong to vmgroup 11 and are on switch port INT8 (that is, host 'esx2'). The HR department VM belongs to vmgroup 21 and is currently visible on switch port INT7.

The **show interface information** command is useful to show the current state of the physical ports on the switch. This command helps track the VLAN to port associations as shown in Example 4-17.

Example 4-17 Displaying the physical port information

```
Router>show interface information
```

Alias	Port	Tag	Type	RMON	Lrn	Fld	PVID	NAME	VLAN(s)
INT1	1	y	Internal	d	e	e	1	INT1	1 4095
INT2	2	y	Internal	d	e	e	1	INT2	1 4095
INT3	3	y	Internal	d	e	e	1	INT3	1 4095
INT4	4	y	Internal	d	e	e	1	INT4	1 4095
INT5	5	y	Internal	d	e	e	1	INT5	1 4095
INT6	6	y	Internal	d	e	e	1	INT6	1 1100 1200 4095
INT7	7	y	Internal	d	e	e	1	INT7	1 1200 4095
INT8	8	y	Internal	d	e	e	1	INT8	1 1100 4095
INT9	9	y	Internal	d	e	e	1	INT9	1 4095
INT10	10	y	Internal	d	e	e	1	INT10	1 4095
INT11	11	y	Internal	d	e	e	1	INT11	1 4095
INT12	12	y	Internal	d	e	e	1	INT12	1 4095
INT13	13	y	Internal	d	e	e	1	INT13	1 4095
INT14	14	y	Internal	d	e	e	1	INT14	1 4095

You can see the actual VLANs configured in the HR-INFRA01 and FIN-WEBAPP01 **vmprofiles**. We can also see their association with a specific physical port on the switch. Switch port INT6 is being used for the external client, so it is configured with all VLANs to allow communication to the VMs being tested.

Consideration: The IBM Virtual Fabric 10gb Switch module for IBM BladeCenter uses VLAN 4095 for the BladeCenter chassis internal management network.

4.3.13 NMotion

Perform a VM migration to observe its effect on the VM-Aware network. Migrate the Finance department WEB VM from esx2 to esx1.

If you are logged on to the CLI interface, the first thing to notice is the log message that displays as the VM migrates (Example 4-18).

Example 4-18 VM Migration notice

```
Oct 24 4:32:44 172.25.160.7 NOTICE vm: 10.0.0.1 moved from port INT8 to port INT7
```

Because the VMready switch is aware of the location of VMs at all times, it tracks these movements in the switch event log. Any changes to the switch configuration are displayed on active CLI sessions and stored in the switch log file.

The **show virt vm** command now displays a slightly different view of VM placement as shown in Example 4-19.

Example 4-19 VM placement updated

IP Address	VMAC Address	Index	Port	VM Group (Profile)
10.0.0.3	00:0c:29:06:54:75	2	IT 11	

10.1.0.2	00:0c:29:06:54:7f	1	IT 11
10.0.0.1	00:50:56:97:5c:81	4	INT7 11 (FIN-WEBAPP01)
10.1.0.1	00:50:56:99:25:73	0	INT7 21 (HR-INFRA01)
10.0.0.2	00:50:56:99:56:ec	3	INT8 11 (FIN-WEBAPP01)

The WEB VM that was previously on port INT8 is now on port INT7 along with **vmgroup** and **vmprofile** information.

The view from **show interface information** changes as well, as shown in Example 4-20.

Example 4-20 Physical port VLAN allocation

```
Router>show interface information
```

Alias	Port	Tag	Type	RMON	Lrn	Fld	PVID	NAME	VLAN(s)
INT1	1	y	Internal	d	e	e	1	INT1	1 4095
INT2	2	y	Internal	d	e	e	1	INT2	1 4095
INT3	3	y	Internal	d	e	e	1	INT3	1 4095
INT4	4	y	Internal	d	e	e	1	INT4	1 4095
INT5	5	y	Internal	d	e	e	1	INT5	1 4095
INT6	6	y	Internal	d	e	e	1	INT6	1 1100 1200 4095
INT7	7	y	Internal	d	e	e	1	INT7	1 1100 1200 4095
INT8	8	y	Internal	d	e	e	1	INT8	1 1100 4095
INT9	9	y	Internal	d	e	e	1	INT9	1 4095
INT10	10	y	Internal	d	e	e	1	INT10	1 4095
INT11	11	y	Internal	d	e	e	1	INT11	1 4095
INT12	12	y	Internal	d	e	e	1	INT12	1 4095
INT13	13	y	Internal	d	e	e	1	INT13	1 4095
INT14	14	y	Internal	d	e	e	1	INT14	1 4095

The VLAN 1100 associated with the WEB VM's profile is automatically displayed in the list of VLANs associated with port INT7 (that is, esx1). VLAN 1100 also remains on INT8, while the DB VM is still present on esx2.

Both these tables update whenever a virtual machine migrates. The result is that network policies dynamically follow virtual machines as they migrate from one physical switch port to another. In a VM-Aware network, the network administrator must define policies only once per VM, regardless of the number of physical ports the VM can migrate across.

If the VM migrates back to esx2, the interfaces are updated again to ensure only the required VLANs are associated with the physical ports (Example 4-21).

Example 4-21 Configuration returns to original state

```
Router>
Oct 24 4:38:58 172.25.160.7 NOTICE vm: 10.0.0.1 moved from port INT7 to port INT8
show interface information
```

Alias	Port	Tag	Type	RMON	Lrn	Fld	PVID	NAME	VLAN(s)
INT1	1	y	Internal	d	e	e	1	INT1	1 4095
INT2	2	y	Internal	d	e	e	1	INT2	1 4095
INT3	3	y	Internal	d	e	e	1	INT3	1 4095
INT4	4	y	Internal	d	e	e	1	INT4	1 4095
INT5	5	y	Internal	d	e	e	1	INT5	1 4095
INT6	6	y	Internal	d	e	e	1	INT6	1 1100 1200 4095

INT7	7	y	Internal	d	e	e	1	INT7	1 1200 4095
INT8	8	y	Internal	d	e	e	1	INT8	1 1100 4095
INT9	9	y	Internal	d	e	e	1	INT9	1 4095
INT10	10	y	Internal	d	e	e	1	INT10	1 4095
INT11	11	y	Internal	d	e	e	1	INT11	1 4095
INT12	12	y	Internal	d	e	e	1	INT12	1 4095
INT13	13	y	Internal	d	e	e	1	INT13	1 4095
INT14	14	y	Internal	d	e	e	1	INT14	1 4095

4.3.14 Testing Policy Compliance

At this stage, test if your configuration of VMready successfully enforces the corporate IT policies defined in 4.2.1, “Example IT policies” on page 49.

Policy 1: Layer 2 Separation

Make sure that Layer 2 is separated as shown in Figure 4-14. Layer 2 separation now exists between the Finance and HR department VMs. The WEB and FILE servers are unable to exchange traffic due to VLAN segregation.

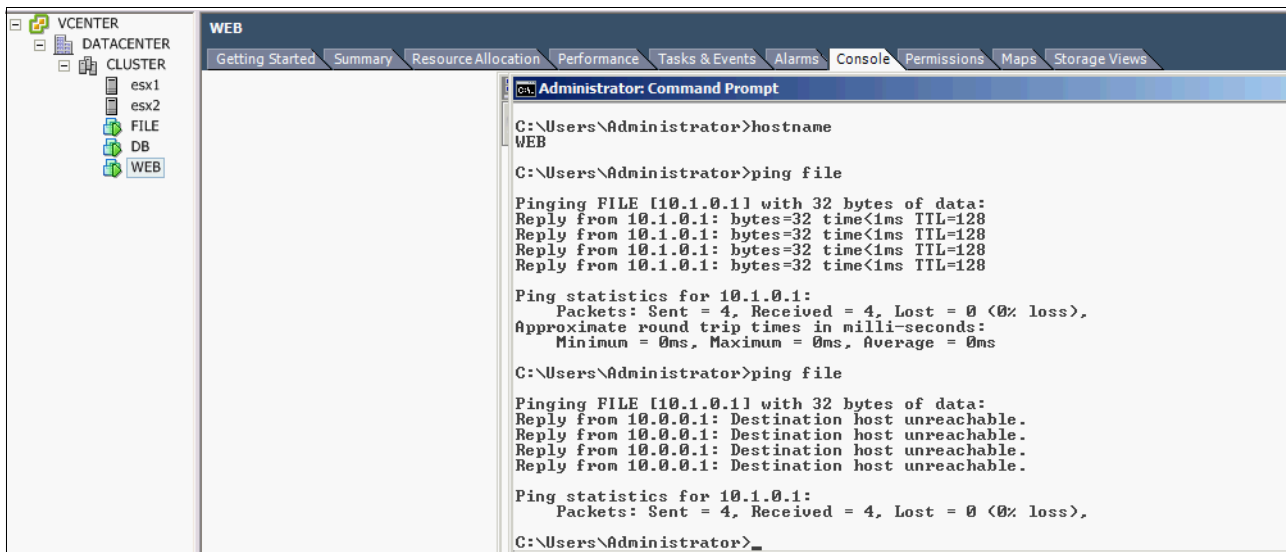


Figure 4-14 Unable to ping from WEB to FILE

Policy 2: No access to default http port

Also, make sure that you have no access to the default port as shown in Figure 4-15.

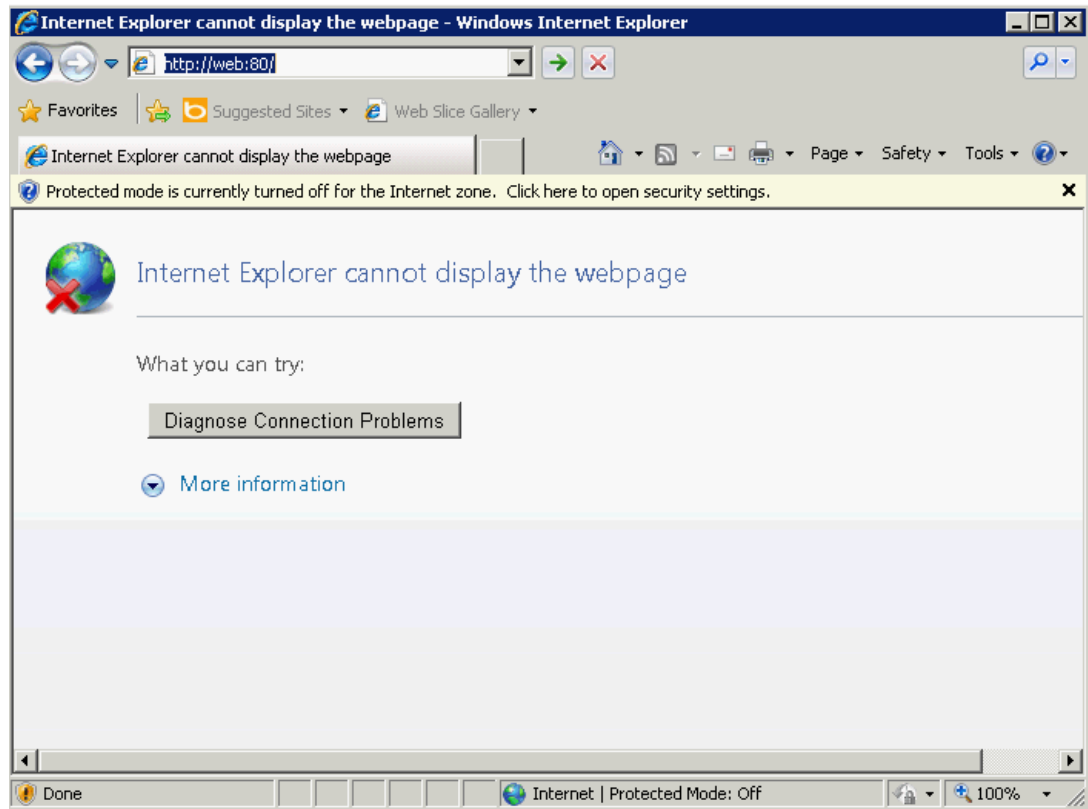


Figure 4-15 Web server no longer reachable on port 80 from Client

The WEB server is no longer accessible on the default HTTP port 80. However, you can still contact the web server on secure port 443 (Figure 4-16).

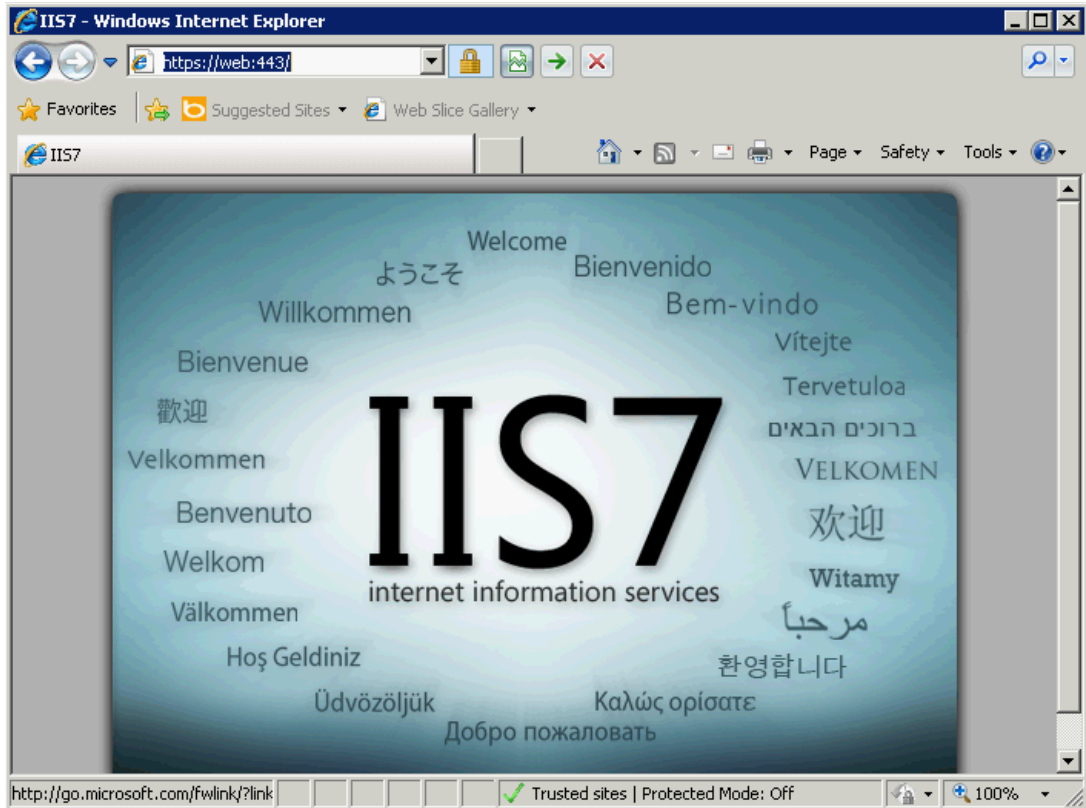


Figure 4-16 WEB server still contactable on HTTPS:443

Policy 3: Limit shared resource usage

Bandwidth limiting has been applied to the HR department FILE server to ensure that traffic is kept below configured limits (Figure 4-17).

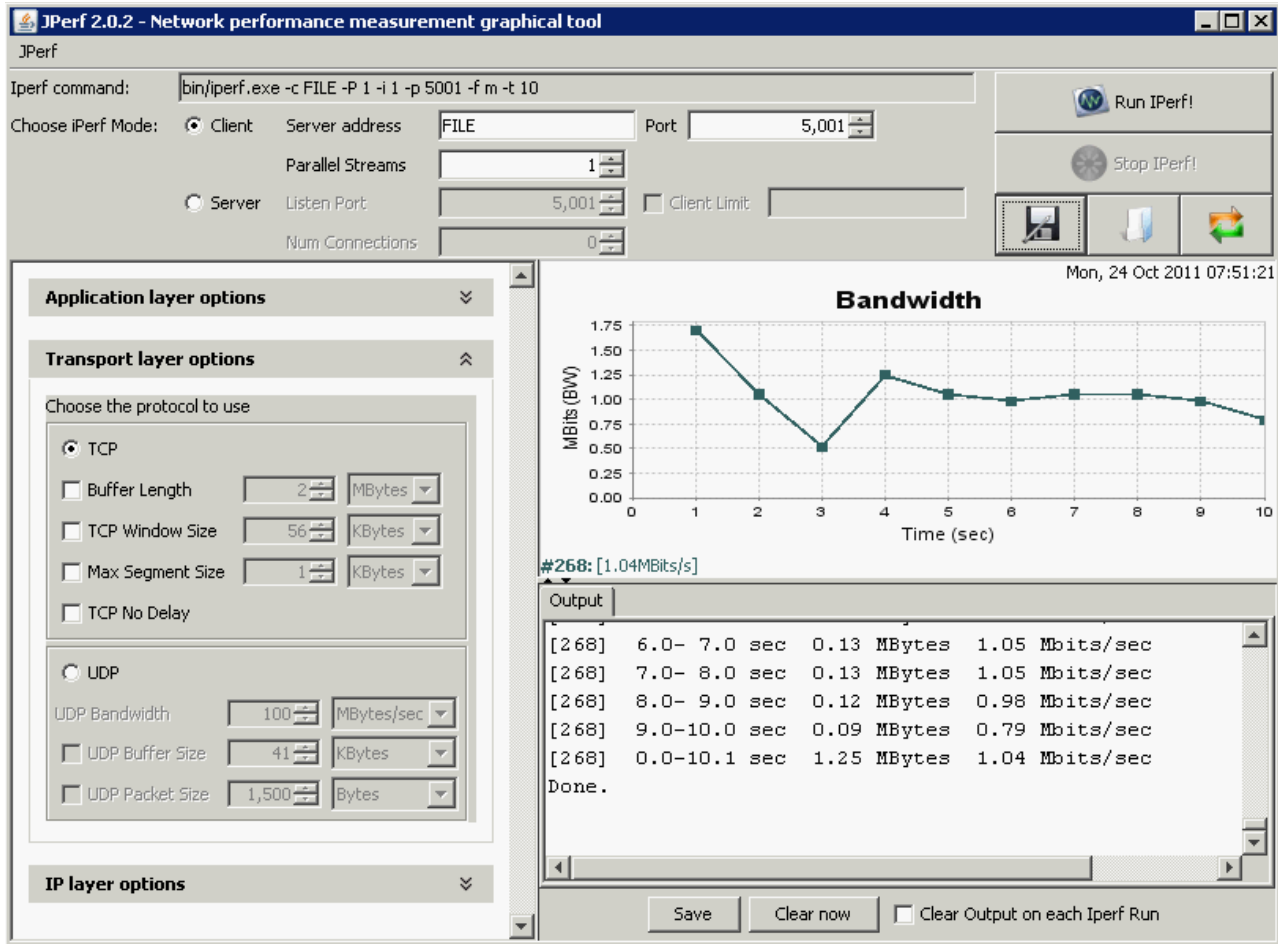


Figure 4-17 JPerf from Client to FILE VM after bandwidth shaping was applied

4.3.15 VMready summary

Implementation of VMready has shown how network administrators can regain control of the network access layer and configure VM-aware networks. This configuration allows network configurations to follow virtual servers whenever they migrate from one physical port to another.

Network policies can now be enforced in a virtual environment from the same interface that traditional access layer network configurations are performed from.

4.4 Implementing VMready with IEEE 802.1Qbg support

As introduced in Chapter 2, “Introducing VMready”, 802.1Qbg/Edge Virtual Bridging (EVB) is an emerging IEEE standard for allowing networks to become VM-aware. VMready with EVB support provides enhanced visibility and control to the physical network. It extends the ability of the physical network to monitor and set network parameters from the physical port level to the virtual machine port level.

This emerging standard provides for a revolutionary approach to implementing virtual machine awareness in the physical network.

A number of component interactions are prescribed by the standard to actively enable VM-awareness within the physical network. For the purposes of implementing EVB support for VMware the following components must be considered:

- ▶ Virtual Station Interface (VSI) type database: This central repository defines network policies that apply to virtual machine network ports.
- ▶ VSI Discovery and Configuration Protocol (VDP) support: This protocol requires the hypervisor and physical network to actively communicate the network state of virtual machines and their specific network requirements. VDP support requires enhancements to the traditional behavior of physical switches and hypervisors to exchange virtual machine networking requirements.
- ▶ Virtual Ethernet Port Aggregator (VEPA) or Virtual Ethernet Bridge (VEB): EVB allows for two possible modes of operation for hypervisor host-internal switches:
 - VEPA: This mode allows the host-internal switch to act as a port aggregator. In this mode, all virtual machine traffic is first passed to the physical switch for processing. The VDP protocol is used to communicate the status of the virtual infrastructure to the physical switch. This approach provides a consistent set of networking capabilities across both physical and virtual networks. The physical network switches must be configured to query a VSI type database and support reflective relay (traffic hairpinning) to allow VEPA mode of operation.
 - VEB: This mode allows virtual machine network traffic to be switched internally to the hypervisor host whenever it is between VMs on the same host. The host-internal switch must still query the VSI type database to apply appropriate network parameters. In addition, the host-internal switch also participates in the VDP protocol with the upstream physical switch. This configuration ensures that network policies are applied uniformly for both host-internal network traffic and traffic that traverses the physical network.

4.4.1 Components required to enable EVB for VMware

The following VMready components enable EVB functionality as set out previously for VMware vSphere 5 environments:

- ▶ IBM Distributed Switch 5000V (5000V): The 5000V provides IT departments with enhanced visibility and control over their virtual machine networks. It is a virtual switch appliance that provides configuration and management interfaces like those of traditional physical switches. This configuration allows network administrators to configure virtual machine networks just as they were able to configure networking for physical hosts.

The 5000V provides a number of advanced features not available in the native vSphere distributed switch. These features include ERSPAN, sFlow, ACLs, QoS, SNMP, RADIUS, TACACS+, LACP, Advanced Teaming, and EVB support. Of particular interest are these features that enable EVB for VMware environments:

- ▶ VSI Type database: The 5000V controller includes an embedded VSI-type database that can be used as the central repository for virtual machine network policies. An alternative database you can use is the VSI-type database included in IBM System Network Element Manager.
- ▶ VDP support: The 5000V enables VDP support for the vSphere 5 hypervisor so the hypervisor can proactively exchange information about virtual machine networking requirements with the physical switch layer.
- ▶ VEPA mode: The 5000V can be enabled for VEPA mode either at an individual virtual machine port level or at a port group level. This configuration allows you to use it both as a host-internal distributed switch for some virtual machines and a VEPA for others. The 5000V provides VDP support for VMs connected to VEPA enabled ports.

Restriction: At the time of writing, the 5000V supports the VDP protocol only for VMs connected to VEPA enabled ports. VMs connected to non-VEPA ports experience network traffic flows akin to those provided by a conventional distributed virtual switch.

Although the 5000V enables EVB support at the vSphere hypervisor level, the physical switch layer also needs to support EVB capabilities. IBM switches that include EVB support are the IBM BNT Virtual Fabric 10 Gb Switch Module for IBM BladeCenter and the IBM RackSwitch G8264.

Table 4-3 shows the software levels used in the example to support EVB functionality in the IBM BNT Virtual Fabric 10 Gb Switch Module.

Table 4-3 Software requirements for VMready with EVB

Component	Software version
IBM System Networking 10 Gb Virtual Fabric Switch Module	IBM Networking OS 6.9.1.0
IBM System Networking Distributed Switch 5000V	Version 1.0.0.1519
ESX Host	ESXi 5.0 build 474610
vCenter Server	vCenter 5.0.0 build 455964

4.4.2 Target scenario

The implementation scenario starts in the same initial state as the VMready implementation in 4.2.3, “Initial configuration” on page 50. To comply to the IT policies, implement the following configuration changes:

- ▶ Connect the Finance WEB and DB servers to the finance web application VLAN: 1100.
- ▶ Connect the HR FILE server to the HR infrastructure VLAN: 1200.
- ▶ Use a group ACL on the Finance web application VLAN to prevent connections to the web server on port 80.
- ▶ Apply a bandwidth policy to the FILE server to restrict the amount of bandwidth it can consume to 1 Mbps.

Figure 4-18 highlights a target configuration that meets the minimum requirements of the example IT policies.

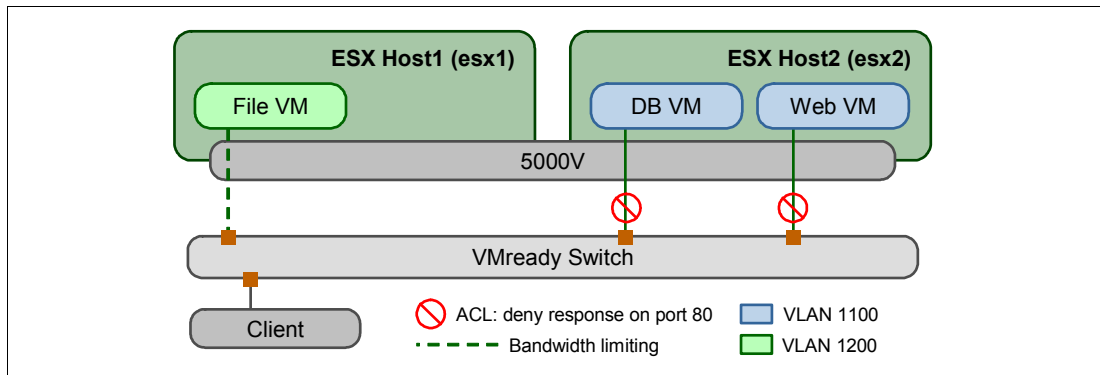


Figure 4-18 Target configuration for VMready with EVB

4.4.3 EVB installation workflow

The steps for Installing VMready with EVB support in a VMWare vSphere environment involves the steps shown in Figure 4-19. These tasks are performed by the system and network administrators. Some tasks are required only for the initial setup, whereas others are repeated for each VM deployment.

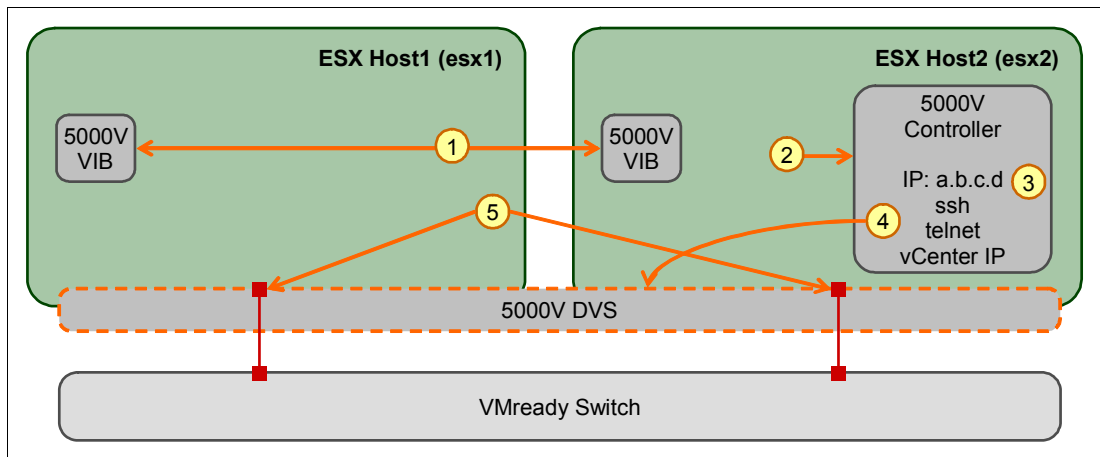


Figure 4-19 EVB installation workflow

Table 4-4 lists the installation steps.

Table 4-4 Initial installation steps

Server Administrator	<ol style="list-style-type: none"> 1. Installs the 5000V vSphere installation bundle (VIB) on the target ESX hosts. 2. Deploys the 5000V controller virtual appliance on an ESX host. You do not need to install the virtual appliance on an ESX host that has uplinks to the Distributed Switch 5000V. It can be installed on any vSphere 4 or 5 ESX host. The controller provides the management interface for the associated instance of the distributed virtual switch.
----------------------	---

Network Administrator	3. Performs the initial configuration of the 5000V controller. This process includes changing the administrator password, setting the IP address, enabling remote management (SSH/telnet), installation of licenses, and configuration of vCenter credentials in the controller.
	4. Creates an instance of the Distributed Switch 5000V in the target vSphere datacenter.
Server Administrator	5. Assigns the hosts and uplinks to the Distributed Switch 5000V.

Requirement: There is a one to one correlation between a 5000V controller appliance and the Distributed Switch 5000V instance. If multiple instances are required, multiple controller appliances must be deployed.

4.4.4 EVB installation steps

The following objects are involved in configuring VMready with EVB support within a VMware vSphere 5.0 environment:

- vib** The 5000V data plane component comes packaged as a vSphere installation bundle that requires installation on each ESX host.
- evbprof** An EVB profile is configured in the physical switch to enable features such as traffic hairpin support (reflective relay). It also enables the VDP protocol on the physical switch ports that connect to the hypervisor.
- vnicprof** A 5000V port group that allows network policies and parameters to be set on a group of virtual ports. This configuration includes enabling VEPA mode within the hypervisor and specifying the associated VSI type ID at the port group level.

The following steps are required to install the components described in 4.4.3, “EVB installation workflow” on page 74.

Step1: Install 5000V offline bundle (VIB)

A number of ways are available to update VMware vSphere hosts with third-party installation packages. These methods include the use of VMware vSphere Update Manager and the vSphere CLI.

The example uses the vSphere CLI. Upload the installation bundle (an archive file) to a shared VMFS volume that is accessible from all target ESX hosts as shown in Figure 4-20.

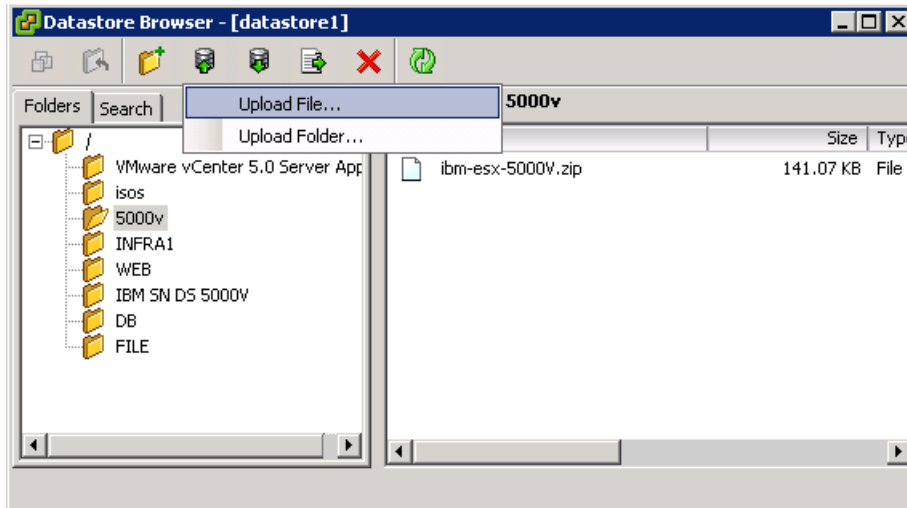


Figure 4-20 Uploading the 5000V offline bundle to shared storage

Install the 5000V installation bundle by using the `esxcli` vCLI command as shown in Example 4-22.

Example 4-22 Deploying the installation bundle to individual ESXi hosts

```
C:\Program Files (x86)\VMware\VMware vSphere CLI>esxcli --server=esx1 software vib install --depot="/vmfs/volumes/datastore1/5000v/Host_Module.zip"
Enter username: root
Enter password:
Installation Result
  Message: The update completed successfully, but the system needs to be rebooted for the changes to be effective.
  Reboot Required: true
  VIBs Installed: IBM_bootbank_ibm-esx-5000V_1.0.0-1461
  VIBs Removed:
  VIBs Skipped:

C:\Program Files (x86)\VMware\VMware vSphere CLI>esxcli --server=esx2 software vib install --depot="/vmfs/volumes/datastore1/5000v/Host_Module.zip"
Enter username: root
Enter password:
Installation Result
  Message: The update completed successfully, but the system needs to be rebooted for the changes to be effective.
  Reboot Required: true
  VIBs Installed: IBM_bootbank_ibm-esx-5000V_1.0.0-1461
  VIBs Removed:
  VIBs Skipped:
```

After deployed, reboot of each host so the changes take effect.

Step 2: Deploy 5000V controller virtual appliance

The 5000V controller ships as an OVA appliance that can be directly imported to a vSphere environment.

Exception: Although only uplinks from ESXi 5.0 hosts can be added to a Distributed Switch 5000V instance, the 5000V controller appliance can run on hosts running ESX 4.0 and later.

Deploy the 5000V controller appliance on host esx2 by selecting the ESX host in the Hosts and Clusters window and clicking **FILE** → **Deploy OVF template**.

Figure 4-21 shows the deployment summary page just before the deployment task.

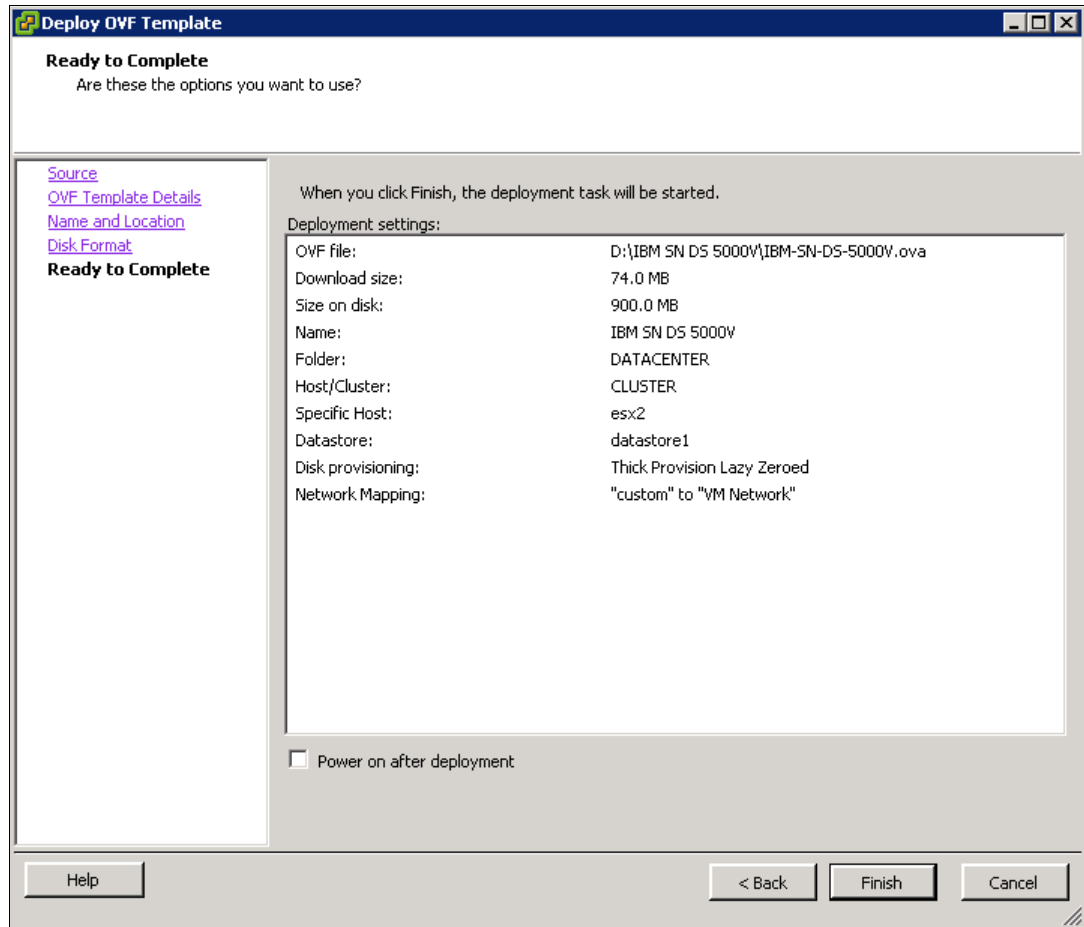


Figure 4-21 Deploying 5000V appliance summary

After uploading, ensure that the 5000V controller virtual machine is connected to a network that has access to the ESX host management IP addresses.

Figure 4-22 shows the network configuration for the 5000V management controller in our environment.

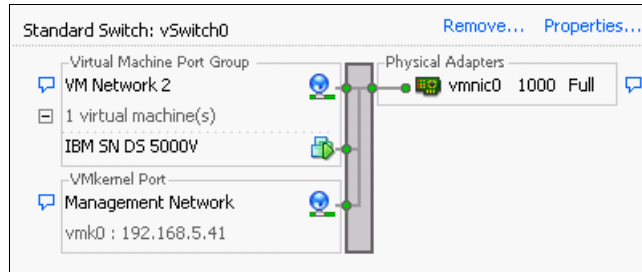


Figure 4-22 5000V controller network

The server administrator powers on the 5000V controller virtual machine to allow the network administrator to continue the configuration process.

Step 3: Initial configuration of 5000V controller

The 5000V controller uses the industry standard CLI for configuration and management tasks. The network administrator connects to the VM console of the 5000V controller through vCenter, and performs the initial configuration of the environment.

The network administrator opens a console window to the 5000V controller virtual machine from the vSphere Client. The administrator then accepts the licensing agreement and logs in with the default password (“admin”). Thereafter, to begin configuring the switch, the network administrator enters privileged mode and starts the configuration terminal shell as shown in Example 4-23.

Attention: CLI commands might vary slightly in subsequent firmware releases due to usability improvements and updates in standards terminology.

Example 4-23 Entering configuration mode

```
5000V> enable
Enable privilege granted.
5000V# configure terminal
Enter configuration commands, one per line. End with Ctrl/Z.
```

Attention: It is essential and imperative that all wanted configuration changes are followed by a command to save the configuration to the controller. This configuration prevents synchronization issues between the 5000V controller and changes made in vCenter if the controller suffers downtime or is restarted. In some circumstances, the controller does not automatically resync with the vCenter configuration. This error can happen when the configuration change requires modification to entities configured outside the controller, such as virtual machine network settings and ESX host uplinks. The **copy running-config active-config** command can be used to save the configuration changes to the active configuration block of the virtual switch.

Change the default administrator password by using the **access user administrator-password** command as shown in Example 4-24.

Example 4-24 Changing the administrator password

```
5000V(config)# access user administrator-password
Changing admin password; validation required.
```

```
Enter current admin password:
Enter new admin password(max 128 characters):
Re-enter new admin password:
New "admin" password accepted.
```

Thereafter, the management IP address can be configured by using the interface **ip-mgmt address**, **netmask**, and **gateway** commands as shown in Example 4-25.

Example 4-25 Configuring the management IP interface

```
5000V(config)# interface ip-mgmt address 192.168.5.50
5000V(config)# interface ip-mgmt netmask 255.255.0.0
5000V(config)# interface ip-mgmt gateway 192.168.0.254
```

Remote administration through SSH and telnet can be enabled with the **ssh enable** and **access telnet enable** commands as shown in Example 4-26.

Example 4-26 Enabling remote administration protocols

```
5000V(config)# ssh enable
5000V(config)# access telnet enable
```

After this step is complete, the administrator can access the 5000V controller remotely just like a traditional managed network switch.

The 5000V controller ships with an initial 8 socket, 60 day evaluation license. This license can be queried with the **show license** command as shown in Example 4-27.

Example 4-27 Showing the evaluation license

```
5000V(config)# show license
Controller ID: fb3be5f715e14b73bd0a8631d31cf458
License Details:

LICENSE           =>      EVALUATION MODE : 8 CPU SOCKETS
USED LICENSE      =>      0
AVAILABLE LICENSES =>      8

59 day(s) until evaluation license expires

HOST LICENSE INFORMATION
-----
```

A license file can be uploaded to a TFTP server and installed on the controller by using the **copy tftp license** command as shown in Example 4-28.

Example 4-28 Installing the license file

```
5000V(config)# copy tftp license
Address or name of remote host: 192.168.0.3
Source file name: 5000V-FOD.key
Confirm download operation (y/n)?: y
Starting download...
```

```
Nov 3 2011 21:32:16 5000V:SYSTEM-INFO: 5000V INFO mgmt: Starting license
download for "license"
/
```

Download completed
Saving to flash completed

Nov 3 2011 21:32:16 5000V:SYSTEM-INFO: 5000V INFO mgmt: license downloaded
from host 192.168.0.3 file "license"
Copied tftp://192.168.0.3/5000V-FOD.key ==> flash:license

Activated New License

Example 4-29 shows the updated view from the **show license** command after a license file is installed that has entitlements for up to 32 sockets.

Example 4-29 After license installation

```
5000V(config)# show license
Controller ID: fb3be5f715e14b73bd0a8631d31cf458
License Details:
```

```
ACTIVATED LICENSES    =>    32
USED LICENSE          =>     0
AVAILABLE LICENSES    =>    32
```

```
HOST LICENSE INFORMATION
-----
```

Step 4: Create an instance of the Distributed Switch 5000V

A single 5000V controller appliance can be associated with a single instance of a Distributed Switch 5000V virtual switch. The first step is to configure vCenter credentials in the 5000V controller appliance.

vCenter credentials are required by the 5000V controller to configure the distributed virtual switch instance. This configuration is achieved by using the **iswitch vcenter <vcenter ip> <username>** command as shown in Example 4-30.

Example 4-30 Configuring the vCenter credentials

```
5000V(config)# iswitch vcenter 192.168.5.12 root
Password:
vCenter-Port (default 443):
5000V(config)# copy running-config active-config
```

Tip: If prompted to change the next boot configuration block from factory to active-config, select **(Y)es**. This setting ensures that the configuration changes persist after switch reboots.

The network administrator can now deploy the instance of the 5000V distributed vSwitch in vCenter by using the **iswitch vds <vds name> <datacenter name>** command (Example 4-31).

Example 4-31 Deploying 5000V VDS and saving the configuration

```
5000V(config)# iswitch vds 5000v-01 DATACENTER
5000V(config)# copy running-config active-config
```

The 5000V VDS is now displayed in vCenter. Click **Home** → **Inventory** → **Networking** to open the window shown in Figure 4-23.

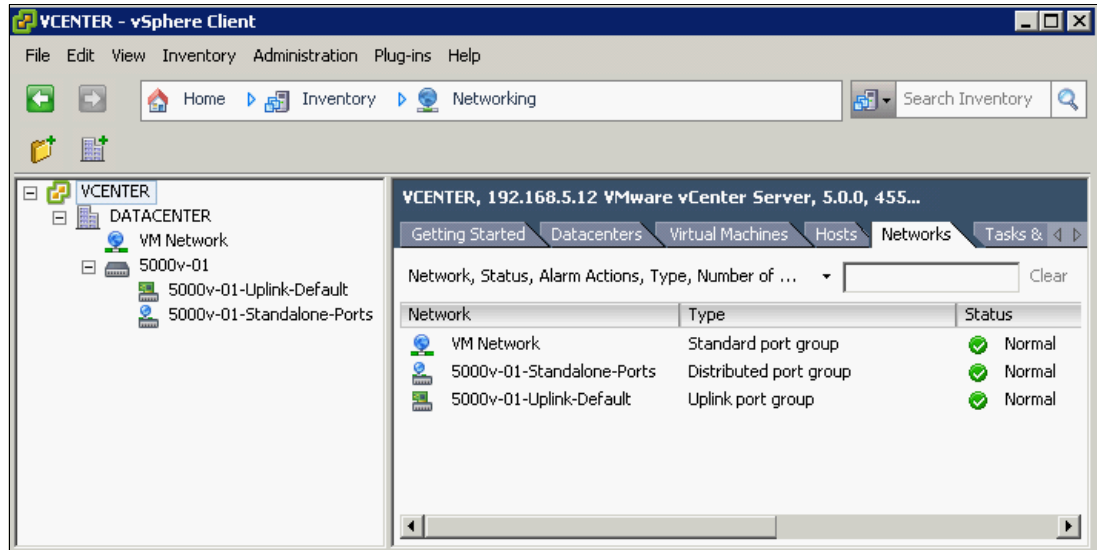


Figure 4-23 Initial state after 5000V VDS deployment

Step 5: Adding hosts and uplinks to the Distributed Switch 5000V

At this stage, the server administrator can add ESX hosts and uplink ports to the Distributed Switch 5000V.

Click **Home** → **Inventory** → **Networking**, then click the **Hosts** tab with an instance of the 5000V VDS selected. Right-click in the empty space under the column headings in this view to add ESX hosts and uplinks to the Distributed Switch 5000V instance. Figure 4-24 shows the instance named 5000v-01.

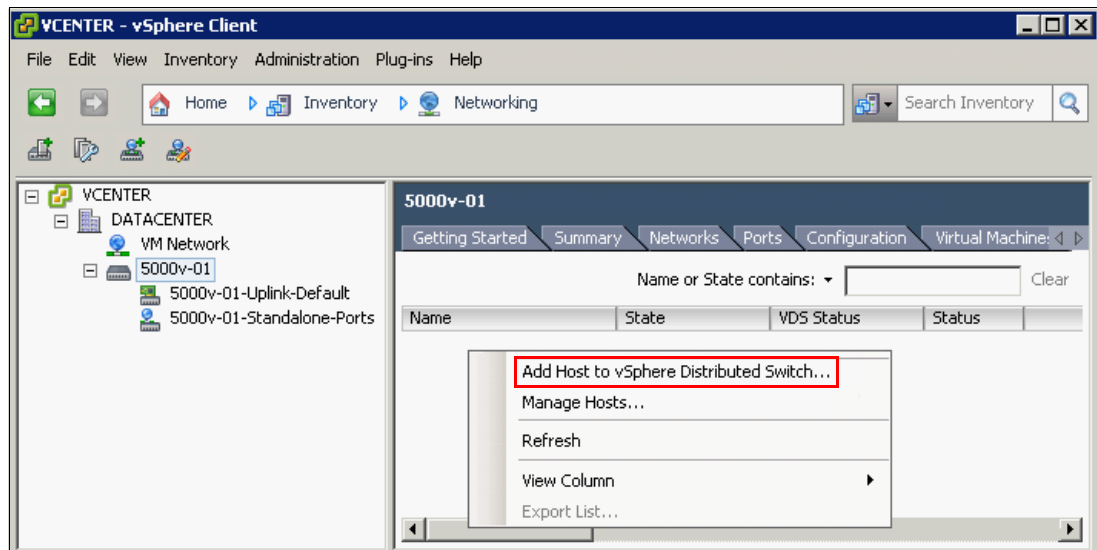


Figure 4-24 Adding hosts to the Distributed Switch 5000V

In the window shown in Figure 4-25, select the wanted hosts and uplinks to allow external connectivity for the Distributed Switch 5000V instance. Add both the esx1 and esx2 hosts and an uplink from each to the default uplink port group.

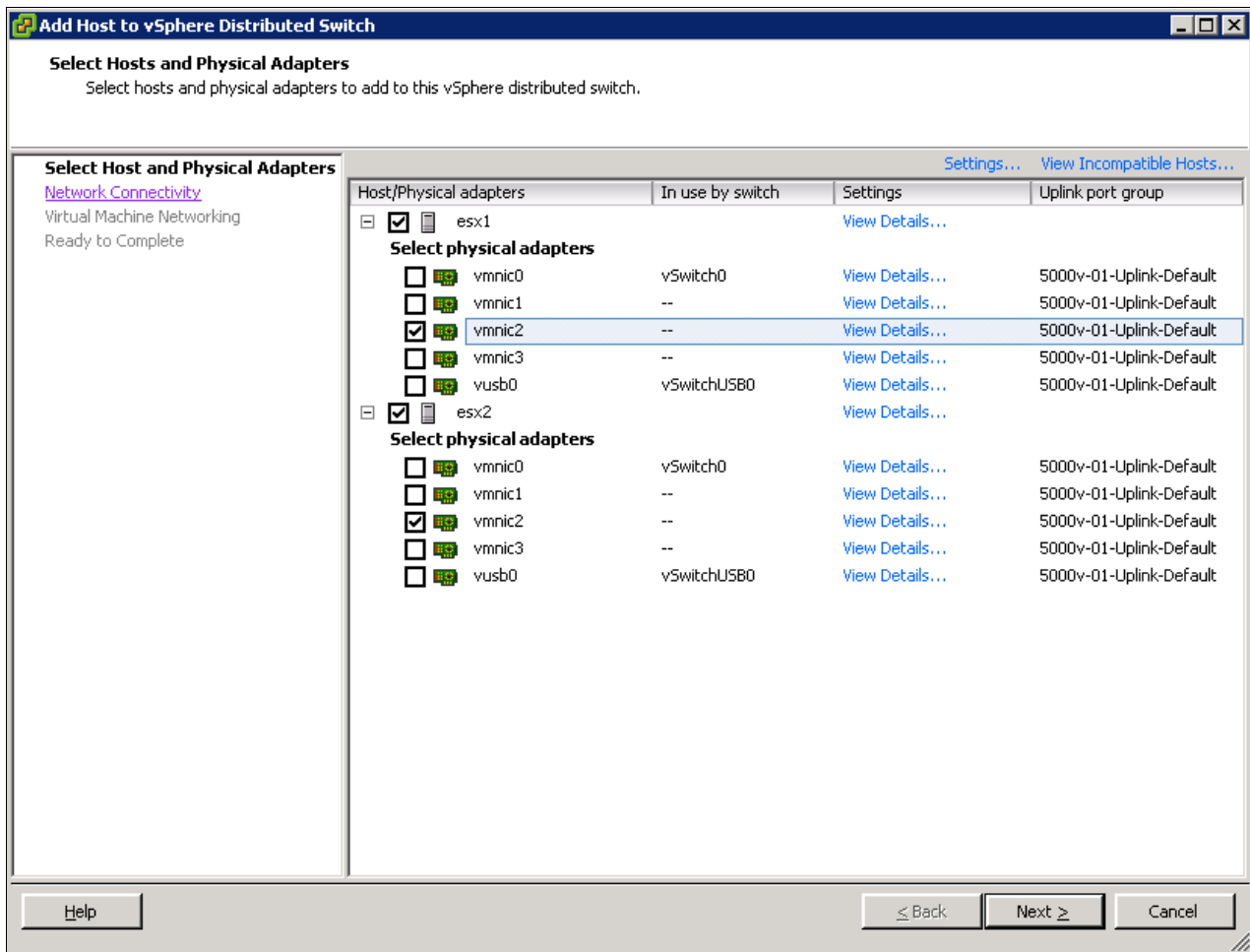


Figure 4-25 Adding uplinks to the Distributed Switch 5000V

After this process is complete, both hosts are displayed under the Hosts tab for the Distributed Switch 5000V instance.

In addition, you can observe the effect of adding hosts through the 5000V controller interface. Example 4-32 shows how the controller licensing has been updated.

Example 4-32 License allocations updated

```
5000V> show license
Controller ID: 8a213960acf6457eb0a40aad16387e5a
License Details:

ACTIVATED LICENSES    =>    32
USED LICENSE          =>     4
AVAILABLE LICENSES   =>    28

HOST LICENSE INFORMATION
-----
```



```
0000 HOST Address: 192.168.5.59
      License ACTIVATED
      Granted Licenses => 02
```

```
0001 HOST Address: 192.168.5.58
      License ACTIVATED
      Granted Licenses => 02
```

The **show iswitch hosts** command also displays information about the hosts currently connected to the Distributed Switch 5000V instance as shown in Example 4-33. This information includes software versions of the virtual switch agent and Data Path Module (DPM) that implement the data plane for the 5000V switch.

Example 4-33 Showing the iswitch hosts

```
5000V(config)# show iswitch hosts
```

```
Hosts Connection Information:
```

IP Address	Age Time	Agent Version	DPM Version
192.168.5.30	3s	1.0.0.1461	1.0.0.1461
192.168.5.41	2s	1.0.0.1461	1.0.0.1461

4.4.5 EVB configuration workflow

Configuring VMready with EVB support to implement the sample IT policies involves tasks performed by the system and network administrators. Some tasks are required only for the initial setup, whereas others are repeated for each VM deployment. To implement the target policies in VMready with EVB support, the respective administrators take the following steps.

Figure 4-26 shows the steps required to configure VMready with EVB.

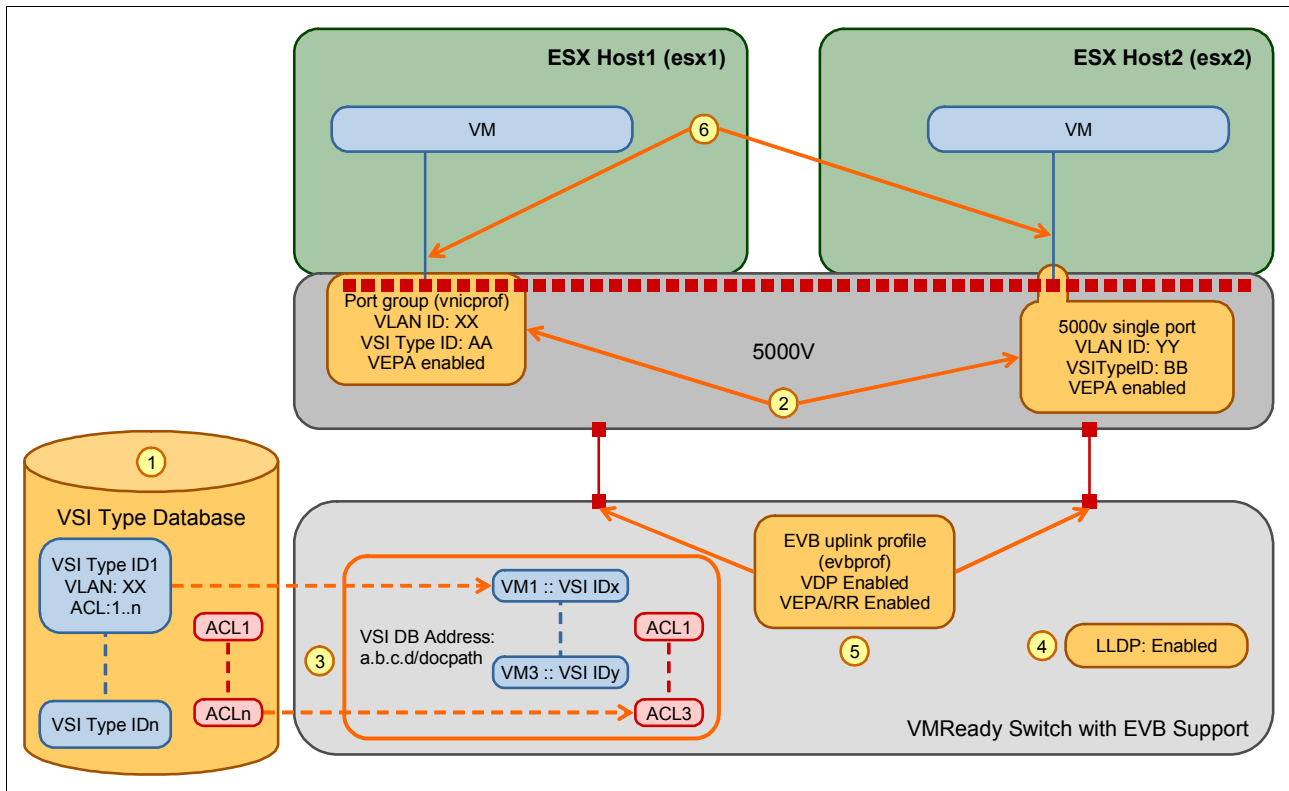


Figure 4-26 Configuring VMready with EVB support

Table 4-5 lists the necessary steps.

Table 4-5 Steps for configuring VMready with EVB support

Network Administrator	1. Creates network policy definitions in the VSI database to cater for virtual machine networking requirements.
	2. Configures VLAN-centric port groups and stand-alone ports with a VSI type ID and native VLAN ID. And also enables VEPA mode for the port group or stand-alone port.
	3. Configures the physical switch to query the VSI type database through the IP address and document path.
	4. Enables Link Layer Discovery Protocol (LLDP) support in the physical switch.
	5. Creates and assigns EVB profiles in the physical VMready switch to enable reflective relay and VDP support on the physical ports connected to the 5000V uplink adapters.

Server Administrator	<p>6. Connects VM network ports to stand-alone ports or port groups on the 5000V that have the target network characteristics for each VM. These characteristics include appropriate VLAN and VSI type ID.</p> <p>At this stage, the physical switch is automatically configured for the networking requirements of each VM connected. This configuration is achieved by an exchange of information between the Distributed Switch 5000V and the physical switch through VDP messages. The 5000V communicates the VM's identity and the required VSI type ID for each VM. The physical VMready switch maintains a mapping between virtual machine network interface controller (NICs) and VSI type IDs. The physical switch then queries the VSI type database and configures VLAN, ACL, QoS, and traffic shaping requirements. It configures them based on the information contained in the VSI type ID definition for each VM.</p>
----------------------	--

4.4.6 EVB configuration steps

This section describes how to implement the policies required to meet the target configuration mentioned in 4.4.2, “Target scenario” on page 73.

Core to achieving policy compliance in an EVB network architecture is the VSI type database. This database serves as a centralized network policy database which defines the network policies available for all virtual machine networks. At the time of writing, there are two available VSI type databases available for the solution. The first is included in the 5000V controller appliance, and the second is in IBM System Networking Element Manager. The example implementation uses the VSI type database included in the 5000V controller.

Step 1: Create network policy definitions

Define two network profiles (VSI type IDs) in your VSI type database. The first is for the Finance web application network (FIN-WEBAPP01:VSI type ID 11). The second is for the HR infrastructure file server (HR-INFRA01:VSI type ID 12).

Create an ACL that forms part of the definition of the FIN-WEBAPP01 VSI type ID. There are a number of ways ACLs can be defined in the physical VMready switch, either through conventional ACL definitions or through VSI types.

Restriction: The implementation of ACLs associated through VSI type definitions currently applies only to traffic *originating* from the virtual machines to which that VSI type ID applies. The source Media Access Control (MAC) address is automatically set in the ACL to that of the individual virtual machine NIC. This is not the case for conventional ACLs that are applied independent of VSI type IDs in the physical IBM System Networking switches.

And even though the ACLs are defined only once in the VSI type database, these ACLs are duplicated automatically in the physical VMready switch and when particular VMs associated with that VSI type ID are displayed on the switch. One ACL definition can create multiple instances to cater for every individual VM to which the VSI type ID applies. This process is NOT the case for conventional group ACLs (like **vmaps**).

If ACLs are not used carefully when associated with VSI type definitions, physical switch ACL maximums might become a limiting factor. This limitation can occur when many VMs are associated with a VSI type ID that has an associated ACL definition.

The commands required to create an ACL to block all traffic that originates from the standard HTTP port 80 are shown in Example 4-34.

Example 4-34 Blocking tcp response from any VM with source port 80 to any destination

```
5000V(config)# access-list ip 128 extended
5000V(config-ext-nacl)# deny tcp any eq 80 any
5000V(config-ext-nacl)# exit
5000V(config)# copy running-config active-config
```

The **vsiman** command set allows administration of the embedded VSI type database in the 5000V controller appliance. Create the VSI Type ID definition in the database for the Finance web application network and name it FIN-WEBAPP01 (Example 4-35). Associate this VSI Type ID with the ACL ID (128) created in Example 4-34. This configuration blocks traffic from port 80 from all VMs that are configured with this VSI Type ID.

Example 4-35 Creating a VSI type ID for Finance

```
5000V(config)# vsiman
5000V(config-vsiman)# typeid 11 version 1
5000V(typeid-version)# name FIN-WEBAPP01
5000V(typeid-version)# vlans 1100
5000V(typeid-version)# acls 128
5000V(typeid-version)# exit
5000V(config-vsiman)# exit
5000V(config)# copy running-config active-config
```

Similarly, create the VSI type ID for the HR infrastructure network and configure bandwidth rate limiting for this VSI type ID (Example 4-36).

Example 4-36 HR infrastructure server VSI type ID

```
5000V(config)# vsiman
5000V(config-vsiman)# typeid 12 version 1
5000V(typeid-version)# name HR-INFRA01
5000V(typeid-version)# vlans 1200
5000V(typeid-version)# qos rx-burst 1024
5000V(typeid-version)# qos rx-cbr 1024
5000V(typeid-version)# qos tx-burst 1024
5000V(typeid-version)# qos tx-cbr 1024
5000V(typeid-version)# exit
5000V(config-vsiman)# exit
5000V(config)# copy running-config active-config
```

Step 2: Configure 5000V port groups and enable VEPA

The network administrator creates VLAN-centric port group definitions that the virtual machines connect to. VEPA mode is also enabled to allow traffic to be forwarded to the physical switch for processing and policy enforcement.

Create the VLANs to use for the native VLAN ID (**pvid**) for the port group and stand-alone port the VMs will be assigned to (Example 4-37).

Example 4-37 Creating and enabling VLANs

```
5000V(config)# vlan 1100
```

```
Nov  4 2011 06:14:12 5000V:VLAN-INFO: INFO mgmt: VLAN number 1100 with name
```

```
"VLAN 1100" created
5000V(config-vlan)# enable
5000V(config-vlan)# exit
5000V(config)# vlan 1200
```

```
Nov 4 2011 06:14:17 5000V:VLAN-INFO: INFO mgmt: VLAN number 1200 with name
"VLAN 1200" created
```

```
5000V(config-vlan)# enable
5000V(config-vlan)# exit
5000V(config)# copy running-config active-config
```

Restriction: The Distributed Switch 5000V tags all outgoing traffic with the pvid of the source port/port group that the traffic originated from. For this reason, the pvid configured for each port group must match a VLAN specified in the corresponding VSI type definition. The physical switch uses this information to validate the association between a VM and its requested VSI type ID. If the VLAN tag differs from the list in the VSI type definition, the physical switch denies association for that VM. The virtual NIC port will show as being in a “blocked” state.

The network administrator now configures port groups and any stand-alone ports for VMs to connect to. Configure them with a **pvid** that exists in the vlan list of the corresponding VSI type ID definition (Example 4-38).

Example 4-38 Creating a VEPA enabled port group

```
5000V(config)# iswitch vnicprof FIN-WEBAPP01_VSI11_1
Ports Allocated: 101-120
To see mapping to vds port ids please execute cli command: show iswitch ports
5000V(config-vprof)# pvid 1100
5000V(config-vprof)# vsitype 11 version 1
5000V(config-vprof)# vepa
5000V(config-vprof)# exit
5000V(config)# copy running-config active-config
```

At this stage, the network administrator can create a VEPA enabled port group for the HR infrastructure file server by using the same procedure. However, to contrast with the previous procedure, use a stand-alone port (port # 12) instead of a port group. Attach the HR FILE VM to the port, and associate VSI type ID 12 with it as shown in Example 4-39.

Example 4-39 Configure stand-alone port for VEPA mode

```
5000V(config)# interface port 12
5000V(config-if)# pvid 1200
Port 12 is an UNTAGGED Port and its PVID is changed to 1200
5000V(config-if)# vsitype 12 version 1
5000V(config-if)# vepa
5000V(config-if)# exit
5000V(config)# copy running-config active-config
```

At this stage, the configuration of the 5000V is complete. The network administrator enables the physical VMready switch to using the newly configured VSI type database and accepting VDP messages on the 5000V uplink ports.

Step 3: Configure physical switch to query the VSI type database

To configure the physical switch for EVB, log on to the physical switch in **ibmnos** CLI mode.

Restriction: At the time of writing, commands to enable EVB functionality in the physical switch were available only in the **ibmnos-cli** mode. CLI commands might vary in subsequent firmware releases due to updates in standards terminology and usability improvements

The first step in configuring EVB support in the physical switch is to disable non-EVB VMready functionality. Disable it by using the **disvmr** command as shown in Example 4-40.

Example 4-40 Disabling non-EVB VMready

```
>> Main# /cfg/virt/disvmr
```

The network administrator can now configure the VSI type database location. The example uses the VSI type database that comes bundled with the 5000V controller appliance (Example 4-41). Alternatively you can use the VSI type database provided in IBM System Networking Element Manager.

Example 4-41 Configuring VSI type DB location

```
>> Virtualization# /cfg/virt/evb/vsidb 1
```

```
-----  
[VSI Type DB 1 Menu]  
managrip - Set VSI DB Manager IP  
port      - Set VSI DB Manager Port  
docpath   - Set VSI DB Document Path  
alltypes  - Set VSI DB Document Path  
interval  - Set VSI DB Update Interval  
cur       - Display current VSI Type configuration  
reset     - Reset VSIDB Info
```

```
>> VSI Type DB 1# managrip 192.168.5.50
```

```
Current IP address: 0.0.0.0  
New pending IP address: 192.168.5.50
```

```
>> VSI Type DB 1# port 80
```

```
Current Port: 80  
New pending Port: 80
```

```
>> VSI Type DB 1# docpath vsitypes
```

```
Current VSI Type DB URI Path:  
New VSI Type DB URI Path: vsitypes
```

```
>> VSI Type DB 1# apply
```

```
>> VSI Type DB 1# save
```

To modify this example to connect to an IBM System Networking Element Manager VSI type database, specify the IP address of the System Networking Element Manager server and the parameters shown in Example 4-42.

Example 4-42 DB parameters

```
>> VSI Type DB 1# port 40080

>> VSI Type DB 1# docpath bhm/rest/vsitypes
```

Step 4: Enable LLDP support

The network administrator then enables LLDP functionality on the physical switch as shown in Example 4-43.

Example 4-43 Enabling global LLDP support

```
>> LLDP configuration# /cfg/12/11dp/on
Current status: OFF
New status:     ON

>> EVB Profile 1# apply
>> EVB Profile 1# save
```

Tip: At this stage, an additional step is required when configuring QVB support in the IBM RackSwitch portfolio. This step involves the commands `/cfg/sys/srvports/add 7` and `/cfg/sys/srvports/add 8`, where ports 7 and 8 are the server uplink ports.

Step 5: Configure and apply EVB profiles

The network administrator creates an EVB profile that can be applied to physical ports on the switch as shown in Example 4-44. This profile governs whether the VDP and reflective relay capabilities are enabled for the physical port.

Example 4-44 Creating an EVB profile

```
>> Main# /cfg/virt/evb/profile 1
-----
[EVB Profile 1 Menu]
  rr      - Enable/Disable VEPA Mode (Reflective Relay Capability)
  vsidisc - Enable/Disable VSI Discovery (ECP and VDP)
  cur     - Display current configuration

>> EVB Profile 1# rr ena
Current Reflective Relay: disabled
New Reflective Relay:     enabled

>> EVB Profile 1# vsidisc ena
Current VSI Discovery: disabled
New VSI Discovery:       enabled

>> EVB Profile 1# apply
>> EVB Profile 1# save
```

The profile is now ready to be applied to physical switch ports that connect to an EVB enabled hypervisor as shown in Example 4-45. In the example, ports 7 and 8 are connected to hosts esx1 and esx2.

Example 4-45 Assigning the VSI profile to host ports

```
>> EVB Profile 1# /cfg/port 7/evbprof 1
Current evb profile ID: 0
New pending evb profile ID: 1
Enable LLDP EVB TLV automatically on port INT7.

>> Port INT7# /cfg/port 8/evbprof 1
Current evb profile ID: 0
New pending evb profile ID: 1
Enable LLDP EVB TLV automatically on port INT8.

>> Port INT8# apply
>> Port INT8# save
```

After the profiles are applied, the network configuration is complete. The server administrator can now provision or migrate virtual machines to the Distributed Switch 5000V port groups and stand-alone ports enabled for VEPA.

Step 6: Connect VMs to VEPA enabled ports

The server administrator is now able to connect virtual machines to the VM-aware network.

This connection can be done in one of two ways: Through VEPA enabled port groups or VEPA enabled stand-alone ports. The scenario connects the WEB and DB servers to the FIN-WEBAPP01_VSI11_1 port group. The FILE server is connected to the virtual switch port 12 (port 11 in the vCenter interface).

Remember: The internal port numbers in the 5000V are an offset to those in the native vCenter port numbering scheme. To facilitate a translation of these port numbers, a column in the Ports view of the Distributed Switch 5000V can be used as a reference. Figure 4-30 on page 94 highlights this view. In addition, the `show iswitch portmap 5000V controller` command can also be used.

Figure 4-27 shows the WEB VM network connection changes to the finance virtual machines networking adapters.

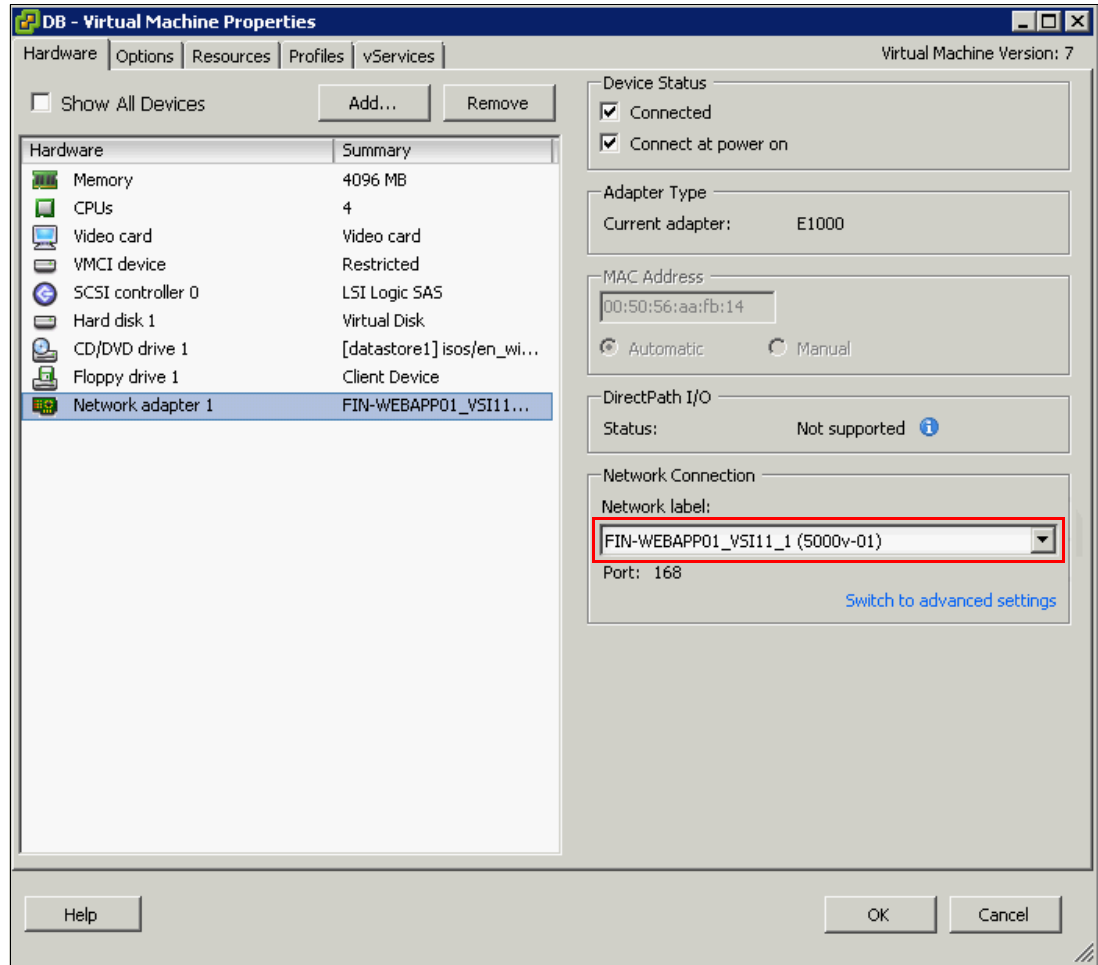


Figure 4-27 WEB VM network connection

Similarly, Figure 4-28 shows the DB VM network connection changes to the finance virtual machines networking adapters.

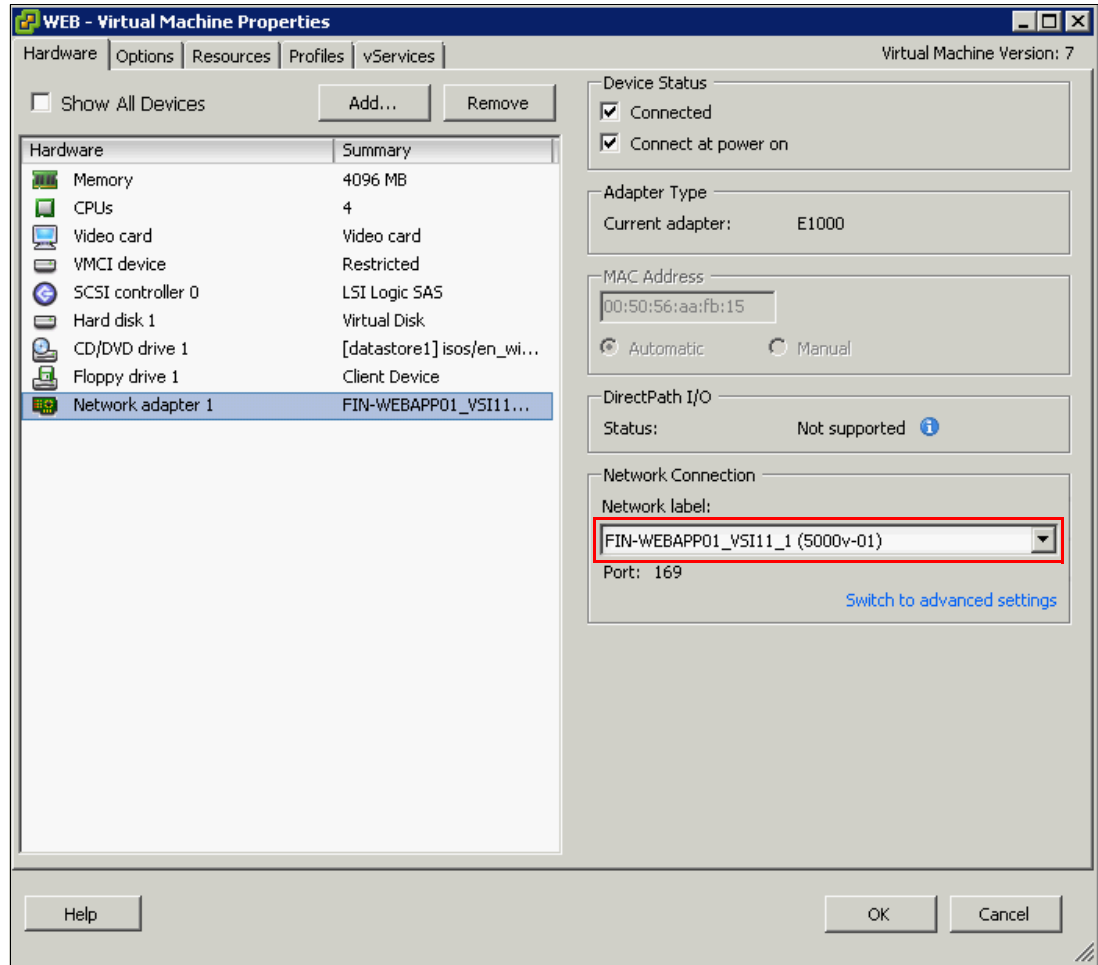


Figure 4-28 DB VM network connection

Figure 4-29 shows the setting used to connect the HR “FILE” virtual machine network adapter to the VEPA enabled stand-alone switch port on the 5000V.

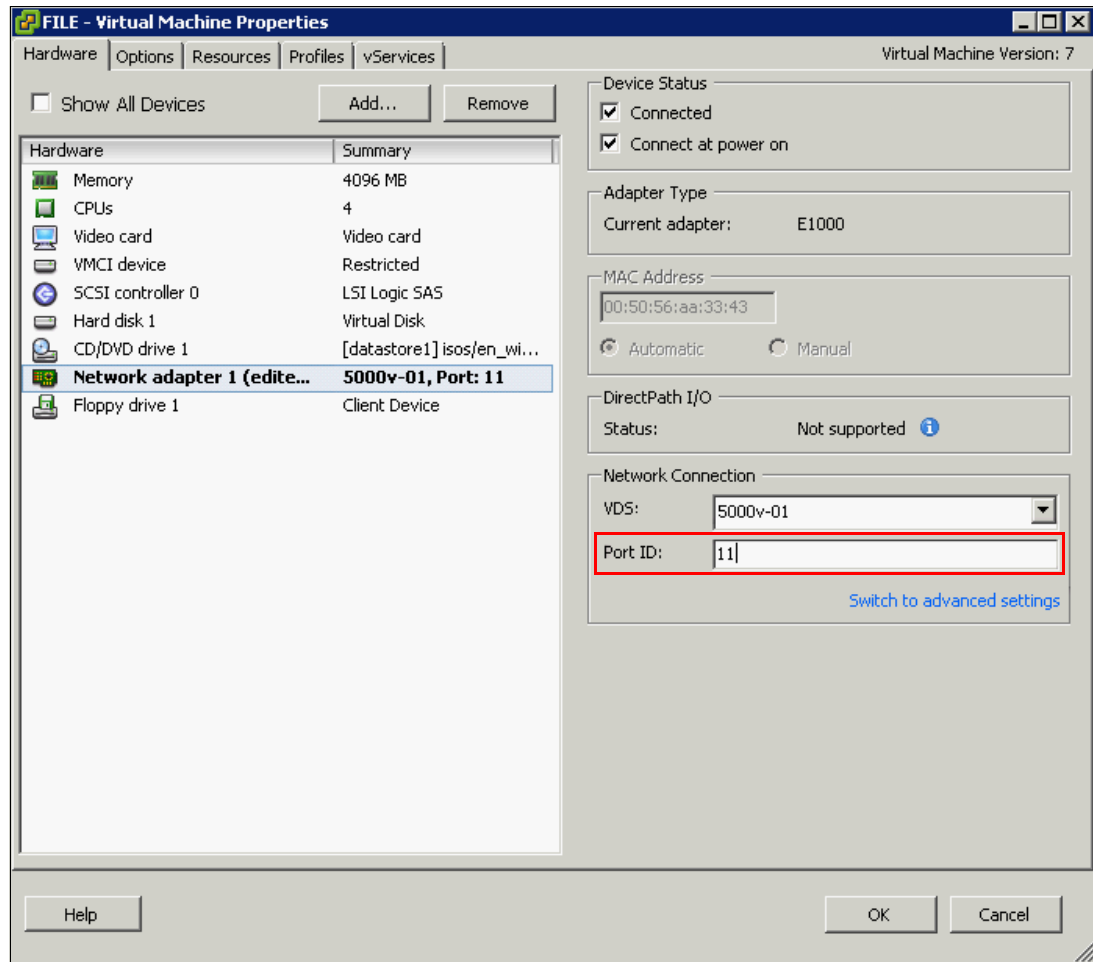


Figure 4-29 FILE VM network connection

4.4.7 Post implementation review

After the VMs are connected to the target Distributed Switch 5000V ports, a series of VDP exchanges between the hypervisor and the physical VMready switch automatically configure the network policies.

The messages in the following examples can be observed in the 5000V controller logs after the virtual machines are connected to the Distributed Switch 5000V. Example 4-46 shows the WEB VM messages.

Example 4-46 WEB VM connection messages on 5000V controller

```
Mar 1 2012 12:24:10 5000V:SYSTEM-INFO: Port 101 connected on Host 192.168.5.41
```

Example 4-47 shows the DB VM messages.

Example 4-47 DB VM connection messages on 5000V controller

```
Mar 1 2012 12:24:22 5000V:SYSTEM-INFO: Port 102 connected on Host 192.168.5.41
```

Example 4-48 shows the FILE VM messages.

Example 4-48 FILE VM connection messages on 5000V

Mar 1 2012 12:28:10 5000V:SYSTEM-INFO: Port 12 connected on Host 192.168.5.30

View virtual port states

To view the state of 5000V virtual ports connected to the virtual machines from vCenter, click **Home** → **Inventory** → **Networking**. Click the **Ports** tab with the instance of the Distributed Switch 5000V selected. Figure 4-30 illustrates the window that is displayed.

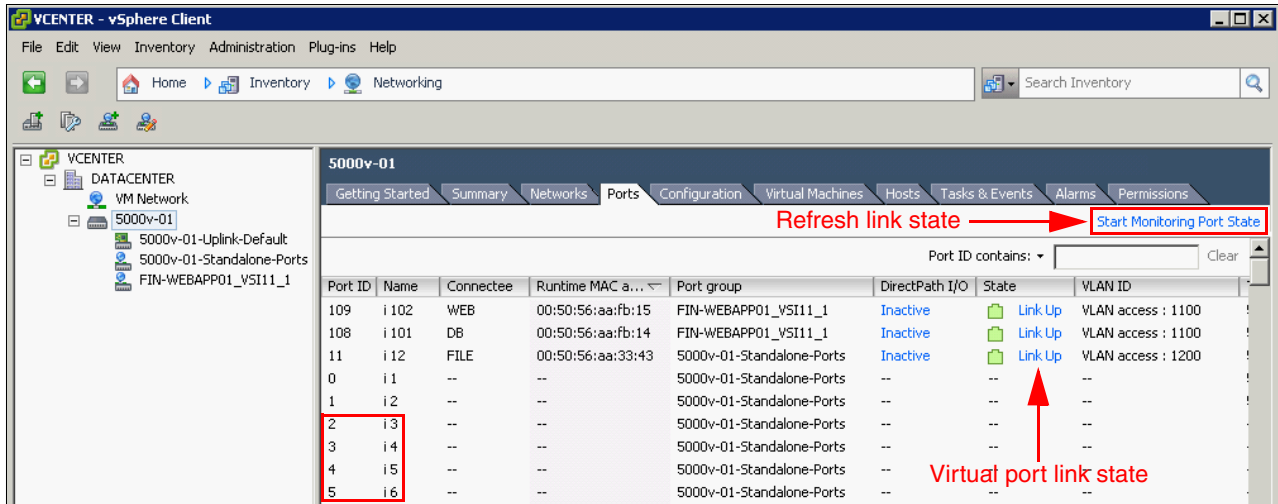


Figure 4-30 Distributed Switch 5000V Ports view

View the mapping of vCenter port numbers (on the left) to the internal port numbers for the 5000V on the right. You can also see the link state of the virtual NICs connected. To refresh the view, click the **Start/Stop monitoring port state** link. Click the link state field for further information about the active configuration of the port. This information is useful in troubleshooting VM connectivity issues.

Figure 4-31 shows the current configuration of the virtual port connected to the WEB VM.

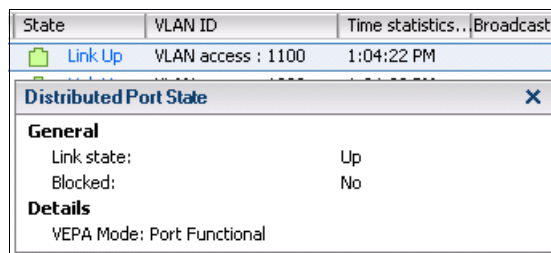


Figure 4-31 5000V port state

Ports can be either in a blocked or non-blocked state. A port can become blocked under a number of circumstances. The states listed in Table 4-6 can be useful in identifying LLDP, VDP, VEPA, or licensing issues.

Table 4-6 5000V port states

Blocked - VDP Disabled	VDP capabilities have not been successfully negotiated with the physical VMready switch.
Blocked - VEPA Disabled	VEPA capabilities have not been successfully negotiated with the physical VMready switch.
Blocked - Not Licensed	The host has not been allocated a license by the 5000V controller.

To check the virtual port state from a CLI interface, issue the **show iswitch ports** command from the 5000V controller as shown in Example 4-49.

Example 4-49 Showing the iswitch ports

```
5000V# show iswitch ports
Port vDs-Port Profile    Connectee    Host          Mac-Address    Status
-----
  1     0 STANDALONE
  2     1 STANDALONE
    ...
    ...
 12    11 STANDALONE  FILE         esx1          00:50:56:aa:33:43 Enabled
    ...
    ...
101   108 FIN-WEBAPP.. DB         esx2          00:50:56:aa:fb:14 Enabled
102   109 FIN-WEBAPP.. WEB        esx2          00:50:56:aa:fb:15 Enabled
```

View VDP associations

As soon as the virtual machines connect to VEPA enabled ports on the 5000V, a VDP exchange occurs between the Distributed Switch 5000V and the VMready physical switch. This results in the physical switch automatically configuring network parameters for virtual machine network traffic on the hypervisor uplink ports.

The following messages can be observed on the physical VMready switch as the VDP exchange occurs for each of the VMs. Example 4-50 shows the WEB VM messages.

Example 4-50 WEB VM connection to IBM BNT 10 Gb VF Switch module

```
Nov 4 12:41:30 192.168.1.17 WARNING vm: Vlan 1100 is created by VM in hardware

Nov 4 12:41:31 192.168.1.17 WARNING vm: Port 8 is tagged and added in vlan 1100 by VM in hardware

Nov 4 12:41:31 192.168.1.17 WARNING vm: VSI Type ID 11 Associated mac 00:50:56:aa:fb:15 on port 8
```

Example 4-51 shows the DB VM messages.

Example 4-51 DB VM connection messages on IBM BNT 10 Gb VF Switch module

```
Nov 4 2011 12:44:51 192.168.1.17 WARNING vm: Port 8 is tagged and added in vlan 1100 by VM in hardware
```

```
Nov 4 2011 12:44:52 192.168.1.17 WARNING vm: VSI Type ID 11 Associated mac
00:50:56:aa:fb:14 on port 8
```

Example 4-52 shows the FILE VM messages.

Example 4-52 FILE VM connection messages on IBM BNT 10 Gb VF Switch module

```
Nov 4 2011 12:51:38 192.168.1.17 WARNING vm: Vlan 1200 is created by VM in
hardware
```

```
Nov 4 2011 12:51:39 192.168.1.17 WARNING vm: Port 7 is tagged and added in vlan
1200 by VM in hardware
```

```
Nov 4 2011 12:51:39 192.168.1.17 WARNING vm: VSI Type ID 12 Associated mac
00:50:56:aa:33:43 on port 7
```

The physical VMready switch receives VDP messages from the Distributed Switch 5000V. These messages request an association of the virtual machine NIC unique VSI instance ID, MAC address, and the VSI type ID that must be applied to that virtual NIC.

In order for the exchange to work, the Distributed Switch 5000V and the physical VMready switch need to have an agreed exchange of capabilities. This exchange is facilitated by LLDP messages. To verify that the physical switch and hypervisor are negotiating LLDP capabilities, use the `/info/12/11dp/port #/tlv/evb` command on the physical VMready switch ports (Example 4-53).

Example 4-53 Query LLDP capabilities

```
>> Main# /info/12/11dp/port 8/tlv/evb
Port Number: 8 (INT8)
      EVB TLV Status:      Local      Remote
      -----
Capability      STD RR RTE ECP VDP  RR RTE ECP VDP
Current Config  RR RTE ECP VDP      RR RTE ECP VDP
VSI Supported   256                    256
VSI Configured  253                    0
RTE             18                      18
```

Example 4-53 shows that the current LLDP-based capabilities match between the local and remote ports. This command is useful for troubleshooting because problems can arise if there is a mismatch of capabilities.

When the physical VMready switch receives an association request from a hypervisor, it runs a number of validation tests before honoring the request. These tests include checking whether the VSI type ID is valid by querying the VSI type database. They also include checking to ensure that the VLAN from which the request was received is contained in the list of VLANs within the requested VSI type ID definition. After an association request succeeds, the switch stores a mapping of virtual NIC MAC addresses and their associated VSI type IDs.

To query the information exchanged, use the `/info/virt/evb/vdp/tlvs` command as shown in Example 4-54. Each of the VM NICs that is associated has a type-length-value (TLV).

Example 4-54 Viewing VDP TLVs

```
>> Optional TLVs# /info/virt/evb/vdp/tlvs
VDP TLVs
Type Length OUI Subtype Request Resp MgrId
-----
127 38 00:17:ef 2 ASSOCIATE 0 0
Type ID : 11
Type Version: 1
Instance ID : 0x49424d30303530353661616662313500
Mac Vlan : 1
Num Entries : 1
VCB Port : 8
VCB Stag : 0
VCB State : 4
VCB timestmp : 674197574
VCB index : 1
Entry : 1
MAC : 00:50:56:aa:fb:15
Vlan : 1100

Type Length OUI Subtype Request Resp MgrId
-----
127 38 00:17:ef 2 ASSOCIATE 0 0
Type ID : 11
Type Version: 1
Instance ID : 0x49424d30303530353661616662313400
Mac Vlan : 1
Num Entries : 1
VCB Port : 8
VCB Stag : 0
VCB State : 4
VCB timestmp : 674197576
VCB index : 2
Entry : 1
MAC : 00:50:56:aa:fb:14
Vlan : 1100

Type Length OUI Subtype Request Resp MgrId
-----
127 38 00:17:ef 2 ASSOCIATE 0 0
Type ID : 12
Type Version: 1
Instance ID : 0x49424d30303530353661613333343300
Mac Vlan : 1
Num Entries : 1
VCB Port : 7
VCB Stag : 0
VCB State : 4
VCB timestmp : 674232216
VCB index : 3
Entry : 1
```

```
MAC      : 00:50:56:aa:33:43
Vlan     : 1200
```

To query the VSI type ID to VM mapping information stored in the physical VMready switch, use the `/info/virt/evb/vdp/vms` command as shown in Example 4-55.

Example 4-55 Querying VSI map

```
>> VSI Information# /info/virt/evb/vdp/vms
VM Associations:
TypeId   MAC                Vlan  Port  TxACL  RxEntry  ACLs
-----
11       00:50:56:aa:fb:15    1100  8     254    51       256
11       00:50:56:aa:fb:14    1100  8     254    51       255
12       00:50:56:aa:33:43    1200  7     254    51       255
```

Viewing the VSI type definitions

To verify that the VSI type ID database is accessible from the physical switch, run the `/info/virt/evb/vdp/vsidb` command as shown in Example 4-56.

Example 4-56 Viewing the VSI database from VMready switch

```
>> VSI Information# /info/virt/evb/vdp/vsidb
Time Since Last Poll: 0 days 0 hours 0 minutes 1 seconds
Time Since Last Update: 0 days 1 hours 54 minutes 15 seconds
```

```
INDEX : 1
```

```
-----
Name           : FIN-WEBAPP01
Type ID        : 11
Version        : 1
Manager ID     : 0
VLAN           : 1100
ACL Index: 1
  SRC MAC      : 00:00:00:00:00:00
  SRC MAC MASK : 00:00:00:00:00:00
  DST MAC      : 00:00:00:00:00:00
  DST MAC MASK : 00:00:00:00:00:00
  VLAN         : 0 (0x000)
  Ether Type   : 0x0800 (IPv4)
  SRC IP       : 0.0.0.0
  SRC IP MASK  : 0.0.0.0
  DST IP       : 0.0.0.0
  DST IP MASK  : 0.0.0.0
  TOS          : 0 (0x00)
  IP Protocol  : 6 (TCP)
  TCP Flags    : 0 (0x00)
  TCP Flags MASK : 0 (0x00)
  L4 SRC Port  : 80 (0x0050)
  L4 SRC Port MASK : 65535 (0xffff)
  L4 DST Port  : 0 (0x0000)
  L4 DST Port MASK : 65535 (0xffff)
  ACL Action   : deny
```

```
INDEX : 2
```

Name	: HR-INFRA01
Type ID	: 12
Version	: 1
Manager ID	: 0
VLAN	: 1200
TxRate	: 1024
TxBurst	: 1024
RxRate	: 1024
RxBurst	: 1024

Verify the availability of the VSI type database configuration at the following URL:

<http://<5000V controller IP address>/vsitypes/>

An XML file should be displayed as shown in Figure 4-32.

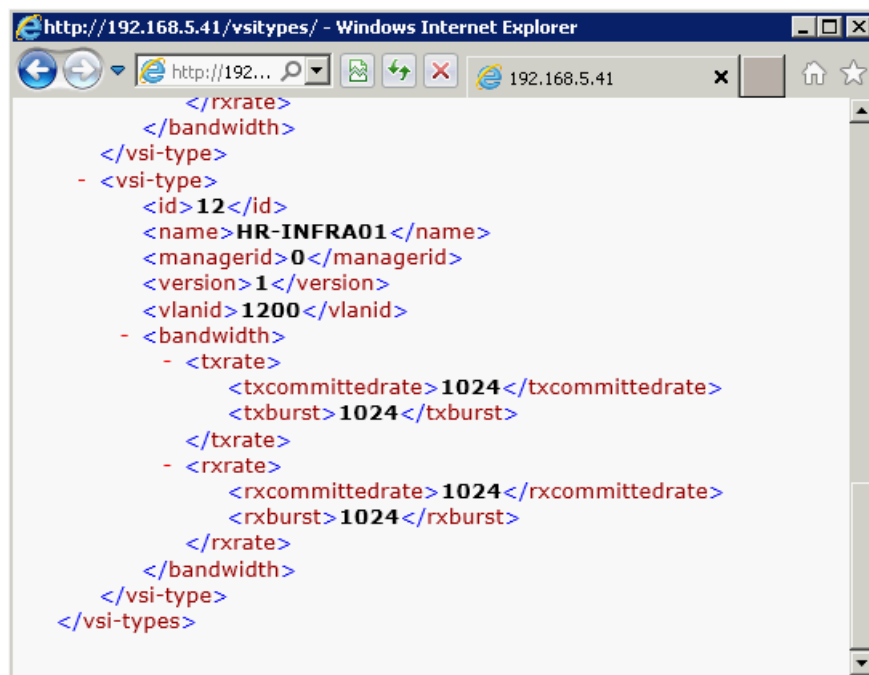


Figure 4-32 VSI type database XML view

4.4.8 NMotion

VMready with EVB support provides virtual machine mobility by design. Using the VDP ensures that the physical switch is made aware of changes to virtual machine network location and parameters. A VDP keep-alive is exchanged between the hypervisor component (Distributed Switch 5000V ports in VEPA mode) and the physical VMready switch. This process ensures that the physical network dynamically associates and disassociates network connectivity and policies for VMs as they progress through their lifecycle.

To observe these dynamic changes in the VM-aware network, migrate the WEB virtual machine from esx2 to esx1. This move corresponds to moving the VMready physical switch from port 8 to port 7.

Example 4-57 shows the migration messages from the perspective of the 5000V controller. The 5000V is a distributed virtual switch spanning multiple hosts. Therefore, it responds to a VM migration by disconnecting the virtual machine from the uplinks of one host and connecting it to the uplinks of another.

Example 4-57 VM migration messages seen on 5000V controller

```
Mar  1 2012 13:06:03 5000V:SYSTEM-INFO: Port 101 disconnected on Host
192.168.5.41

Mar  1 2012 13:06:03 5000V:SYSTEM-INFO: Port 101 connected on Host 192.168.5.30
```

From the perspective of the physical VMready switch, a number of interactions occur with the hypervisor host through the Distributed Switch 5000V. First, the hypervisor sends a dissociate VDP message to the physical switch. This message informs the switch that the VM no longer requires network connectivity on the uplinks of the host from which the VM is migrating. An associate message is then sent through the uplinks of the destination host to inform the physical switch of these settings:

- ▶ The new location of the virtual machine
- ▶ The required network parameters represented by a VSI type ID

In response, the switch validates the VDP association. It then ensures that the appropriate network parameters are applied to the uplinks of the destination host for the VM's connectivity requirements. Example 4-58 shows that the VMready switch has automatically added VLAN 1100 to port 8 before completing the association.

Example 4-58 VM migration messages seen on physical VMready switch

```
May 17 22:31:13 192.168.1.17 WARNING vm: VSI Type ID 11 Dissociated mac
00:50:56:aa:fb:15 from port 8

May 17 22:31:13 192.168.1.17 WARNING vm: Port 7 is tagged and added in vlan 1100
by VM

May 17 22:31:13 192.168.1.17 WARNING vm: VSI Type ID 11 Associated mac
00:50:56:aa:fb:15 on port 7
```

The VSI mapping table is also updated to reflect the new location of the VM as shown in Example 4-59.

Example 4-59 VSI Information

```
>> VSI Information# /info/virt/evb/vdp/vms
VM Associations:
TypeId   MAC                               Vlan  Port  TxACL  RxEntry  ACLs
-----  -
11       00:50:56:aa:fb:15                1100  7     254    51       256
11       00:50:56:aa:fb:14                1100  8     254    51       255
12       00:50:56:aa:33:43                1200  7     254    51       256
```

If you migrate the WEB VM back to esx2, the messages shown in Example 4-60 are seen in the physical VMready switch.

Example 4-60 VM migration messages seen on physical VMready switch

```
May 17 22:36:34 192.168.1.17 WARNING vm: VSI Type ID 11 Dissociated mac
00:50:56:aa:fb:15 from port 7
```

May 17 22:36:34 192.168.1.17 WARNING vm: Port 7 is removed from vlan 1100 by VM
May 17 22:36:34 192.168.1.17 WARNING vm: VSI Type ID 11 Associated mac
00:50:56:aa:fb:15 on port 8

The VM-aware network has removed port 8 from VLAN 1100, so there are no virtual machines on esx1 that require connectivity to this VLAN. Only the FILE VM remains on this host, and it is part of the HR infrastructure VLAN 1200.

4.4.9 Testing policy compliance

Test if the configuration of VMready with EVB successfully enforces the corporate IT policies defined in 4.2.1, “Example IT policies” on page 49.

Policy 1: Layer 2 Separation

A ping test from the finance WEB server to the HR FILE server now fails (Figure 4-33). Layer 2 separation now exists between the Finance and HR department VMs. The WEB and FILE servers are unable to exchange traffic due to VLAN segregation.

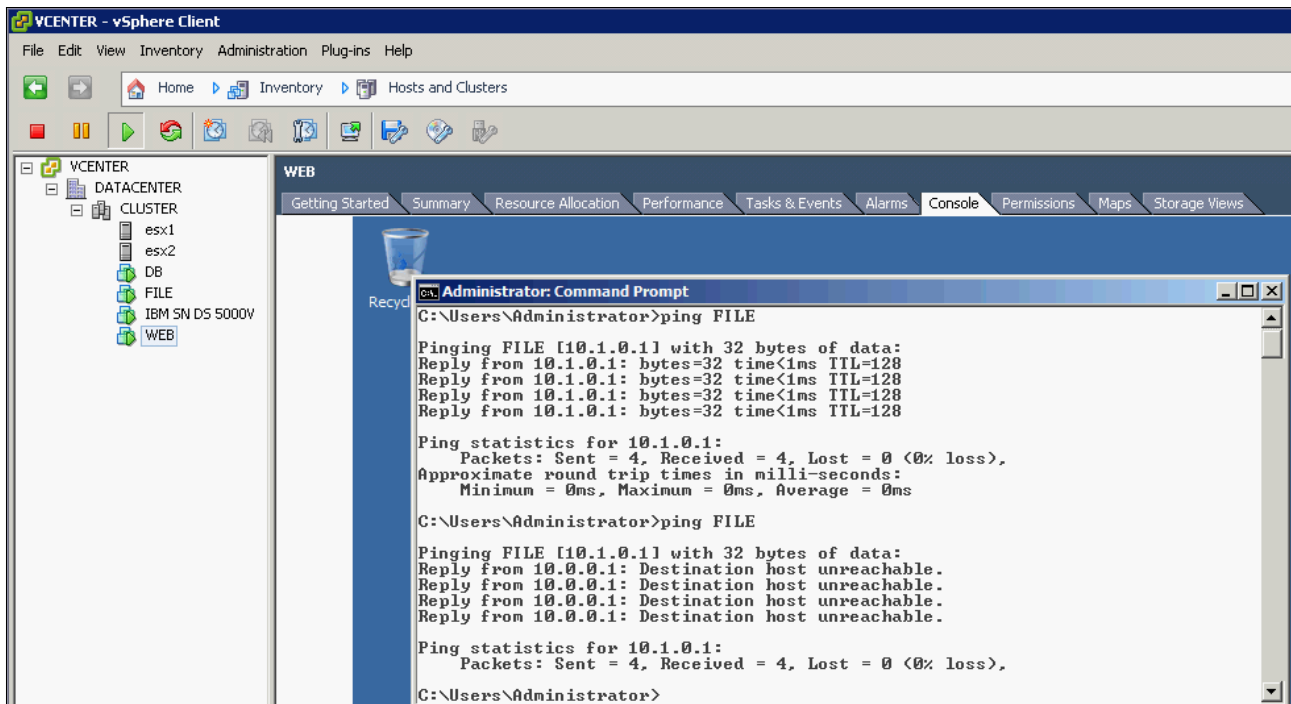


Figure 4-33 Unable to ping from WEB to FILE

Policy 2: No access to default HTTP port

You can no longer reach the default HTTP port by using a web browser as shown in Figure 4-34.

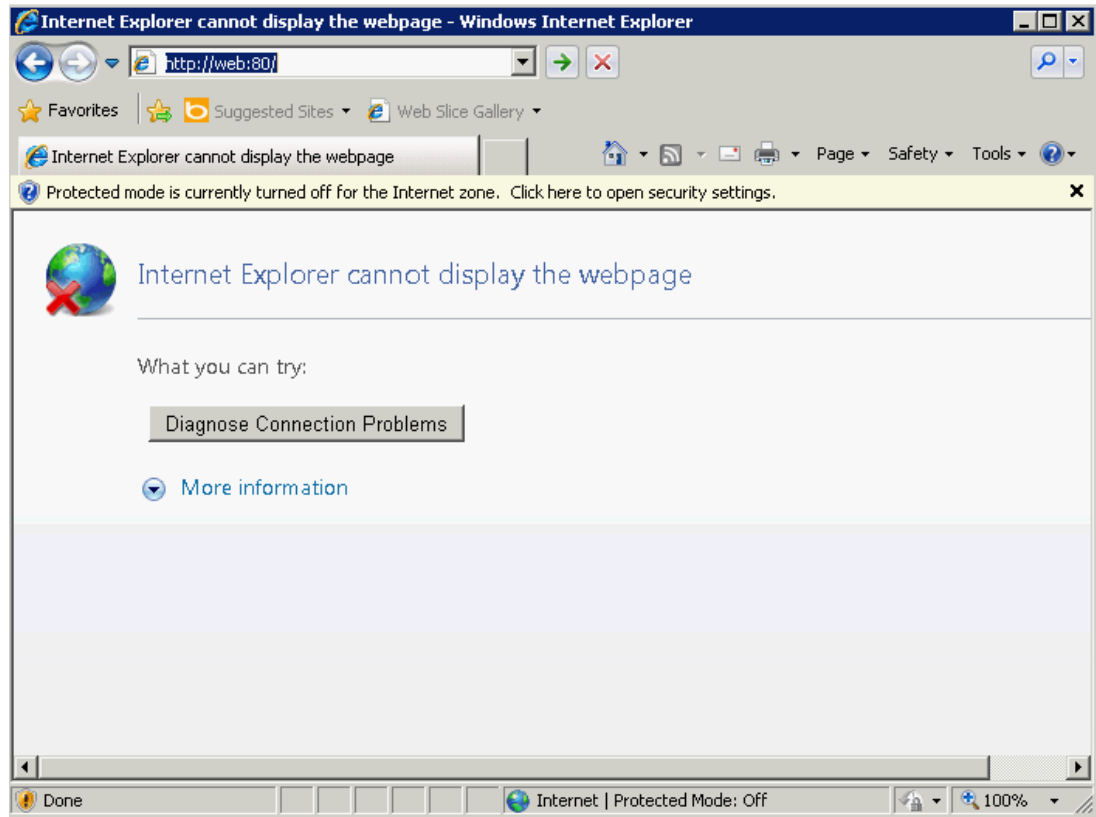


Figure 4-34 Web server no longer reachable on port 80 from Client

The WEB server is no longer accessible on the default HTTP port 80. However you can still contact the web server on secure port 443 as shown in Figure 4-35.

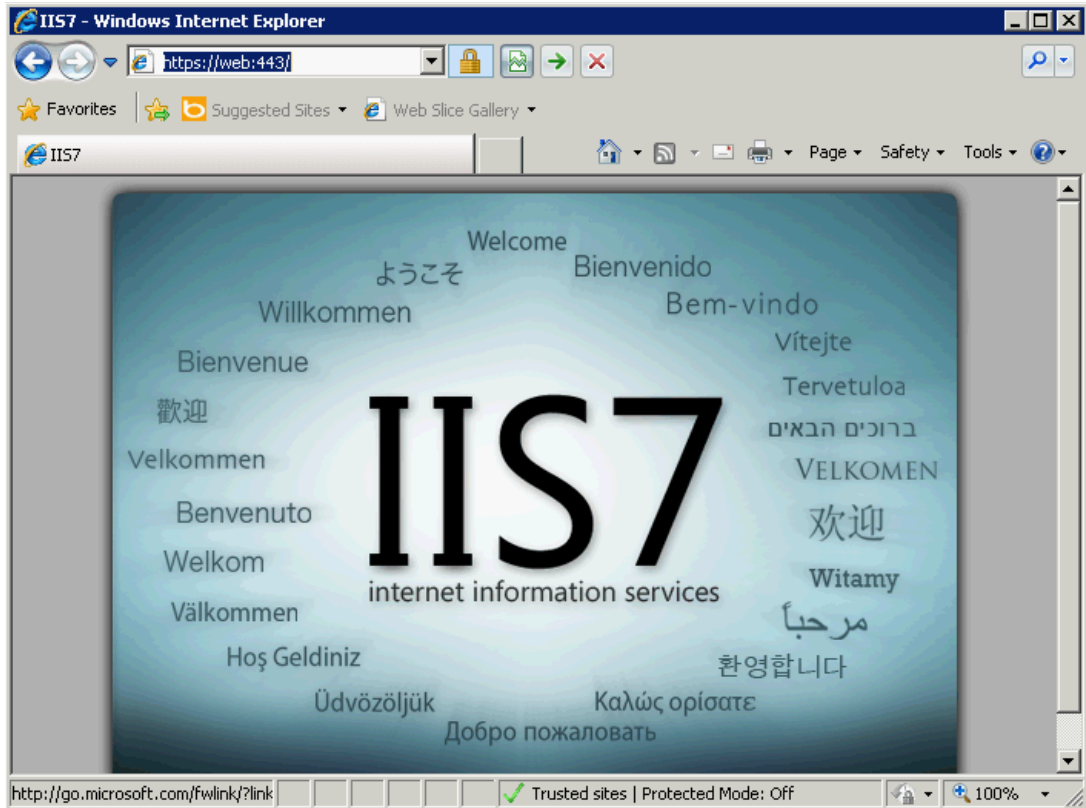


Figure 4-35 WEB server still reachable on HTTPS:443

Policy 3: Limit shared resource usage

Bandwidth limiting has been applied to the HR department FILE server to ensure that traffic is kept below configured limits as shown in Figure 4-36.

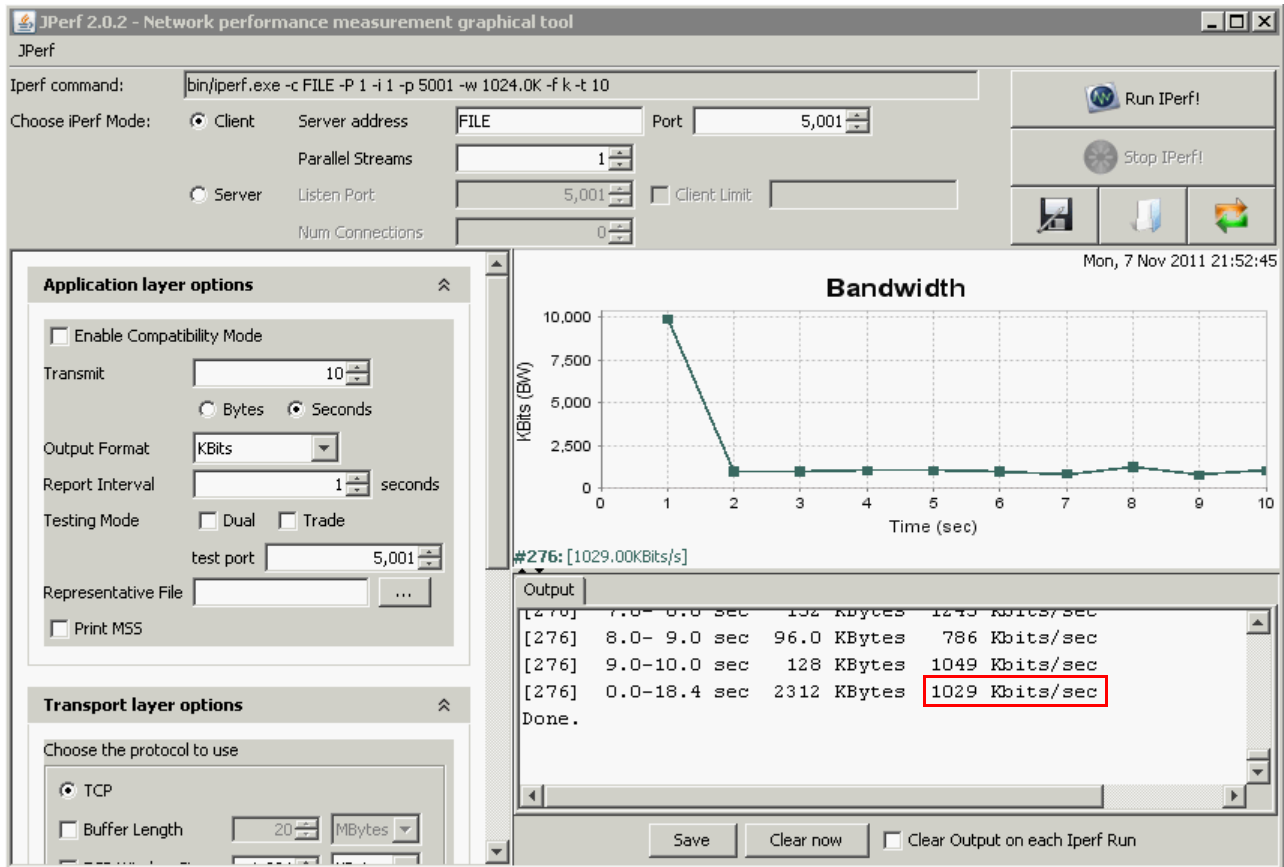


Figure 4-36 JPerf from Client to FILE after bandwidth shaping to 1 Mbps



Implementing VMready to support PowerVM

This chapter describes the steps involved in implementing VMready for PowerVM environments. This chapter includes the following sections:

- ▶ Overview of VMready for PowerVM
- ▶ Initial configuration
- ▶ Initial tests
- ▶ Enabling VMready and implementing network policies
- ▶ Validating the network policies
- ▶ Validating Nmotion

5.1 Overview of VMready for PowerVM

IBM VMready enables the network to automatically discover PowerVM partitions, and apply network settings to them. It also moves those network settings as the partitions are moved.

5.1.1 A little about PowerVM networking

Networking can be implemented in several ways with PowerVM. Partitions can be directly connected to a physical adapter, to a logical port of a Host Ethernet Adapter (HEA) or to a Shared Ethernet Adapter (SEA). With the first and the second choice, partitions are linked to the hardware that host them and Partition Mobility is not an option. To allow partitions to migrate from one server to another, they must be connected to Shared Ethernet Adapters. The SEA is a virtual switch managed by the VIO server. The SEA can be bridged to a HEA, which enables the partitions to communicate outside the VIO Server.

5.1.2 The example system

As shown in Figure 5-1, the implementation and example scenario is based on two host systems with three partitions (FILE, WEB, and DB), and a client system. The client system is used to demonstrate the impact of implementing network policies and to show that those policies are preserved as partitions are moved.

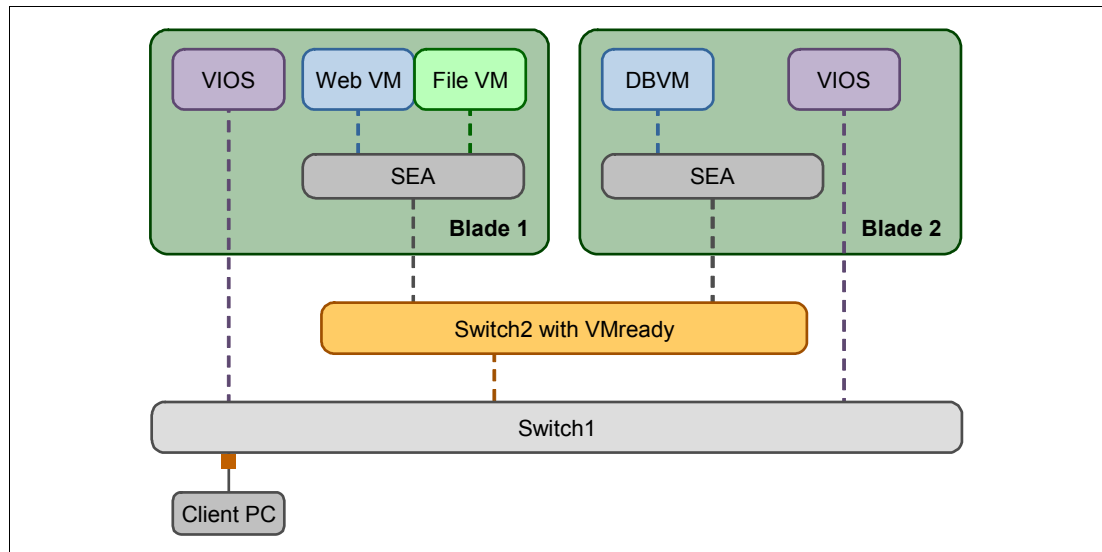


Figure 5-1 Example system configuration

5.1.3 The example scenario

To demonstrate VMready functions, a scenario in which a company provides several web applications to its employees is used. Network administrators want to enforce the security by blocking all non-encrypted connections on port 80 on the web servers, but allowing encrypted connections on port 443. In addition, a file server is available for the employees to download documents. Network administrators want to limit the bandwidth of this server.

Implementing this scenario involves the following steps:

1. Describe the hardware and software environment of the scenario
2. Run some initial tests to highlight the weakness of the environment

3. Enable VMready and implement network policies
4. Demonstrate the impact of these network policies
5. Migrate the partitions and show that their network settings automatically moved with them.

5.2 Initial configuration

This section describes the initial configuration of the example system.

5.2.1 Hardware

The environment is composed of one BladeCenter H. Two dedicated PS702 blade servers are positioned in bay #7 and bay #8 of the chassis as shown in Figure 5-2. The chassis has two IBM BNT 1/10 Gb Uplink Ethernet Switch Modules in bay #1 and bay #2.

The screenshot shows the IBM BladeCenter H Advanced Management Module interface. The main content area displays a table titled "Blades" with the following data:

Bay	Status	Name	Pwr	Owner**		cKVM*	I/O Compatibility	WOL*	Local Control	
				KVM	MT*				Pwr	KVM
1	On	Blade1-HS21	On				OK	On	✓	✓
2	On	Blade2-HS20	On				OK	On	✓	✓
3	On	Blade3-LS21	On				OK	On	✓	✓
4	On	Blade4-LS21	On				OK	On	✓	✓
5	Off	Blade5-HS22	Off				OK	On	✓	✓
6	Off	Blade6-HS22	Off				OK	On	✓	✓
7	On	PowerVM1	On				OK	N/A	✓	✓
8	On	PowerVM2	On				OK	N/A	✓	✓
9	Off	BLADE#01	Off				OK	On	✓	✓
10	On	BLADE#01	On	✓	✓		OK	On	✓	✓
11	Off	BLADE#01	Off				OK	On	✓	✓
12	Off	BLADE#01	Off				OK	On	✓	✓
13	Off	BLADE#01	Off				OK	On	✓	✓
14	Off	BLADE#01	Off				OK	On	✓	✓

Blades 7 and 8 are circled in red in the original image. The interface also includes a sidebar with navigation options like "System Status", "Blade Tasks", and "I/O Module Tasks".

Figure 5-2 BladeCenter overview

5.2.2 VIO Server

Set up one Virtual I/O Server (VIOS) on each of PS702 blades. The version of the VIOS used for this example is 2.2.0.0. The VIOS are installed on the local disk of the blades. Use the Integrated Virtualization Manager (IVM), which is provided with the VIOS. The VIOS are named VIO1 and VIO2.

5.2.3 Shared storage

All the logical partitions (LPARs) created during this scenario use an external SAN storage (Figure 5-3). To demonstrate the migration of the network policy with Live Partition Mobility, directly map your Fibre Channel LUNs to the LPARs by using physical volumes. Each LPAR has a dedicated LUN on the storage device.

Figure 5-3 List of the physical disks

5.2.4 Network

Use the first IBM BNT 1/10 Gb Switch Module in bay #1 to handle the VIOS network traffic. The second IBM BNT 1/10 Gb Switch Module in bay #2 supports the LPARs network traffic. As shown in Figure 5-4, connect all LPARs on one single SEA. Two VLANs (#720 and #721) are enabled on it.

```
$ lsmmap -all -net
SVEA Physloc
-----
ent0 U8406.71Y.06D935A-V1-C3-T1

SEA NO SHARED ETHERNET ADAPTER FOUND

SVEA Physloc
-----
ent1 U8406.71Y.06D935A-V1-C4-T1

SEA NO SHARED ETHERNET ADAPTER FOUND

SVEA Physloc
-----
ent2 U8406.71Y.06D935A-V1-C5-T1

SEA NO SHARED ETHERNET ADAPTER FOUND

SVEA Physloc
-----
ent3 U8406.71Y.06D935A-V1-C6-T1

SEA NO SHARED ETHERNET ADAPTER FOUND

SVEA Physloc
-----
ent6 U8406.71Y.06D935A-V1-C12-T1

SEA ent7
Backing device ent5
Status Available
Physloc U78A5.001.WIHA0D1-P1-T5

$ entstat -all ent7 | grep -i vlan
VLAN Ids :
  VLAN Extract: False
  VLAN tagged filtering mode: Filter according to VLAN permit array
Max number of VLAN IDs per HEA port: 20
Invalid VLAN ID Packets: 0
Port VLAN ID: 10
VLAN Tag IDs: 721 720

$
```

Figure 5-4 Shared Ethernet Adapters and VLAN overview

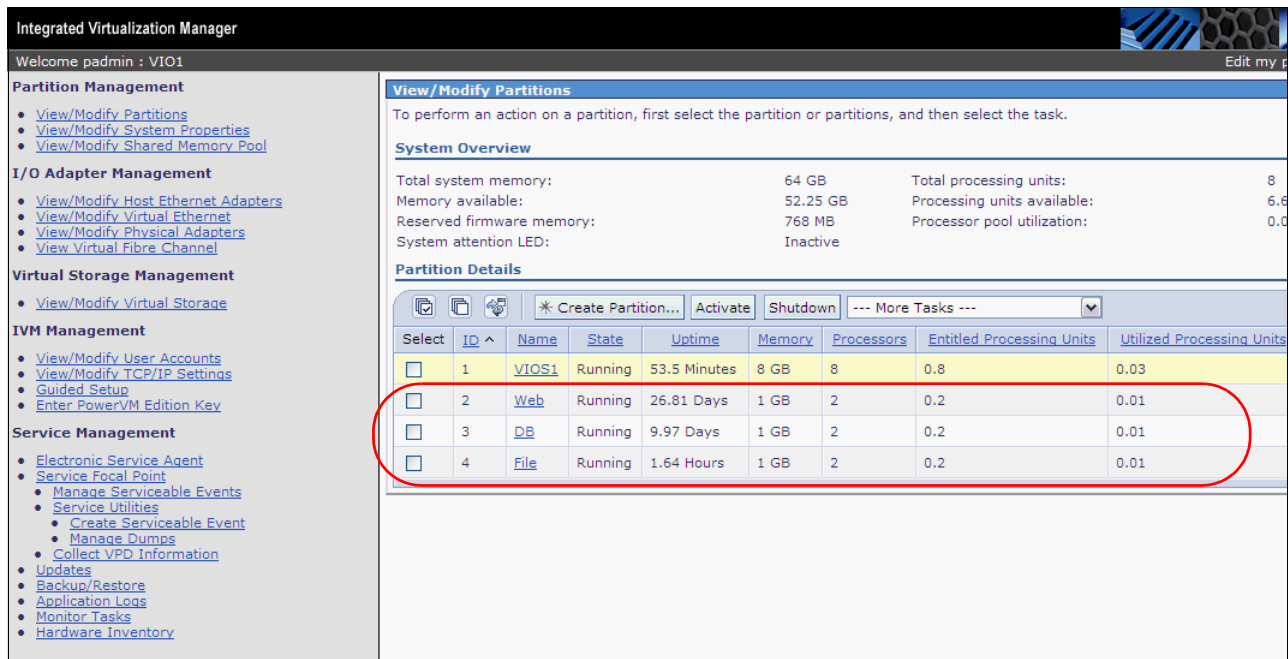
5.2.5 LPARs

Provision three partitions that run AIX 7.1, as shown in Figure 5-5. The LPAR names are Web, DB, and File. At the beginning of the scenario Web and File are hosted on PowerVM1, and DB is hosted on PowerVM2. Web is configured with an HTTP daemon (HTTPd) server that listens on port 80 and port 443.

The LPAR IP address configurations are as follows:

- ▶ Web server: 10.6.220.241 / 255.255.255.0 (VLAN #720)
- ▶ DB server: 10.6.220.242 / 255.255.255.0 (VLAN #720)
- ▶ File server: 10.6.221.243 / 255.255.255.0 (VLAN #721)

The following scenarios use a Windows client to simulate an employee workstation. It is external to the chassis. For the demonstrations, this client has one network interface on VLAN #720 and the other on VLAN #721, as shown in Figure 5-5.



The screenshot shows the Integrated Virtualization Manager interface. The left sidebar contains navigation menus for Partition Management, I/O Adapter Management, Virtual Storage Management, IBM Management, and Service Management. The main window displays the 'View/Modify Partitions' section, which includes a 'System Overview' and a 'Partition Details' table. The 'Partition Details' table lists four partitions: VIOS1, Web, DB, and File. The Web, DB, and File partitions are circled in red.

Select	ID ^	Name	State	Uptime	Memory	Processors	Entitled Processing Units	Utilized Processing Units
<input type="checkbox"/>	1	VIOS1	Running	53.5 Minutes	8 GB	8	0.8	0.03
<input type="checkbox"/>	2	Web	Running	26.81 Days	1 GB	2	0.2	0.01
<input type="checkbox"/>	3	DB	Running	9.97 Days	1 GB	2	0.2	0.01
<input type="checkbox"/>	4	File	Running	1.64 Hours	1 GB	2	0.2	0.01

Figure 5-5 LPARs overview

5.2.6 Switch configuration

The IBM BNT 1/10 Gb Switch Module in bay #2 has the following configuration at the beginning of the scenario (Figure 5-6 on page 111):

- ▶ VLAN #720 and #721 exist.
- ▶ INT7, INT8, and EXT7 are members of these VLANs.
- ▶ Tagging is enabled on the ports.
- ▶ The PVID is set to VLAN #2 as a garbage VLAN for non-tagged traffic.
- ▶ The client can communicate with the LPARs through EXT7, which is connected to the top of rack switch.

```

Router#sho run
Current configuration:
!
version "6.8.2"
switch-type "IBM Networking OS 1/10Gb Uplink Ethernet Switch Module for IBM
BladeCenter"
!
interface port INT7
    pvid 2
    exit
!
interface port INT8
    pvid 2
    exit
!
interface port EXT7
    tagging
    pvid 2
    exit
!
vlan 2
    enable
    name "VLAN 2"
    member INT7-INT8,EXT7
!
!
vlan 720
    enable
    name "VLAN 720"
    member INT7-INT8,EXT7
!
!
vlan 721
    enable
    name "VLAN 721"
    member INT7-INT8,EXT7

```

Figure 5-6 Extract of the configuration of the IBM BNT 1/10 Switch Module in bay #2

5.3 Initial tests

Run the following tests on the configuration from 5.2, “Initial configuration” on page 107:

- ▶ Open an internet connection from the client workstation to the Web server. Use an HTTP non-secure connection on port 80 and an HTTPS secure connection on port 443.
- ▶ Generate a network traffic from the client workstation to the File server to estimate the bandwidth between the two systems.

5.3.1 Web connection test

This test shows that the client workstation can open a connection on the Web server by using a secure or a non-secure connection. Figure 5-7 shows that HTTP connections are enabled from the client.



Figure 5-7 Home page on Web server using a non-secure connection (HTTP on port 80)

Figure 5-8 shows that HTTPS connections are enabled from the client.



Figure 5-8 Home page on Web server using a secure connection (HTTPS on port 443)

5.3.2 Bandwidth test

This test measures the current bandwidth between the client workstation and the File server. Use IPERF/JPERF. Set up the File server as IPERF client and the client workstation as JPerf server. As shown in Figure 5-9, the current bandwidth reaches 850Mbits/s.

Restriction: The IBM BNT 1/10 Gb Switch Module can limit only the outgoing traffic from the blades.

```
# iperf -c 10.6.221.235 -P 1 -i 1 -p 5001 -f m -t 10 -T 1
-----
Client connecting to 10.6.221.235, TCP port 5001
TCP window size: 0.25 MByte (default)
-----
[ 3] local 10.6.221.243 port 33579 connected with 10.6.221.235 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec   112 MBytes  936 Mbits/sec
[ 3] 1.0- 2.0 sec   112 MBytes  935 Mbits/sec
[ 3] 2.0- 3.0 sec   112 MBytes  937 Mbits/sec
[ 3] 3.0- 4.0 sec   111 MBytes  931 Mbits/sec
[ 3] 4.0- 5.0 sec   112 MBytes  937 Mbits/sec
[ 3] 5.0- 6.0 sec   102 MBytes  855 Mbits/sec
[ 3] 6.0- 7.0 sec   99.0 MBytes 830 Mbits/sec
[ 3] 7.0- 8.0 sec   99.2 MBytes 833 Mbits/sec
[ 3] 8.0- 9.0 sec   99.5 MBytes 835 Mbits/sec
[ 3] 9.0-10.0 sec   97.1 MBytes 815 Mbits/sec
[ 3] 0.0-10.0 sec  1054 MBytes 885 Mbits/sec
```

Figure 5-9 IPERF test

5.4 Enabling VMready and implementing network policies

The following sections show how VMready helps configure the network policies addressed in 5.1.3, “The example scenario” on page 106. Implementing the policies involves these steps:

1. Enabling VMready
2. Enabling locally administered MAC addresses
3. Creating VM Groups
4. Applying access control lists
5. Applying traffic shaping

5.4.1 Enabling VMready

To configure VMready on the switch and then implement policies, first enable the VMready functionality.

Enabling from the web interface

To enable VMready, connect to the IBM BNT Virtual Fabric 10 Gb in bay #2 with a web browser. Click the **Configuration** tab, then click **IBM Networking OS 1/10Gb Uplink Ethernet Switch Module** → **Virtualization** → **Virtual Machine** → **Global**. Then select

Enabled in the **Virtual Machine Groups** list. To validate your choice, click **Submit**, then apply and save the configuration (Figure 5-10).

Remember: When configuring, you need to click **Submit** to commit a change, **Apply** to activate it, and then **Save** to keep your configuration active after a switch reboot.

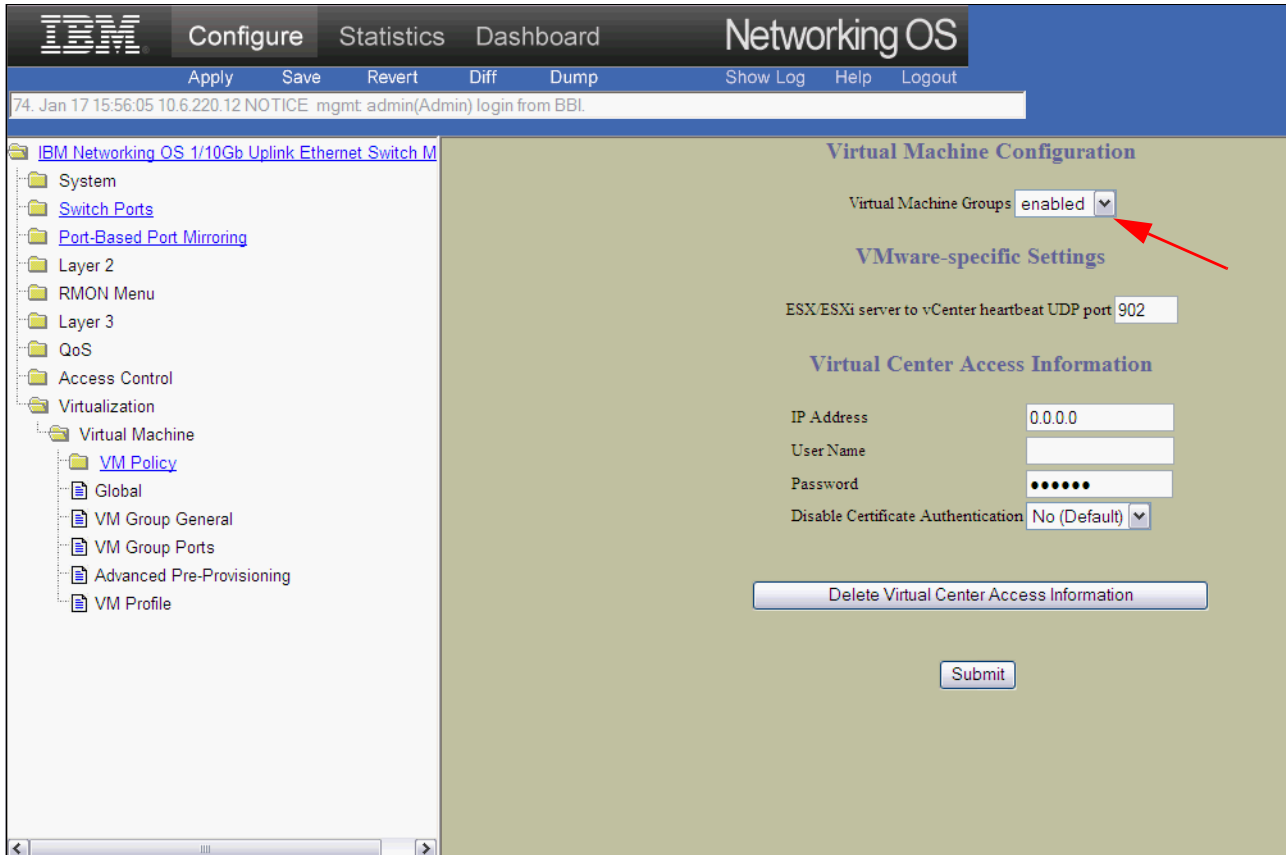


Figure 5-10 Enabling VMready

Enabling from the CLI

To enable VMready, connect to the IBM BNT 1/10 Gb Switch Module in bay #2 of the chassis. Figure 5-11 shows how the `virt enable` command enables VMready from the CLI by using Industry Standard CLI (ISCLI).

```
Router>enable
Enable privilege granted.

Router#conf terminal
Enter configuration commands, one per line. End with Ctrl/Z.

Router(config)#virt enable
Router(config)#^Z

Router#write
```

Figure 5-11 Enabling VMready

5.4.2 Enabling locally administered MAC addresses

At LPAR creation, the VIO server generates one unique MAC address for each LPAR Ethernet adapter. These Media Access Control (MAC) addresses are not part of any registered ranges of addresses (MAC OUI). Also, the BNT switches do not take these private MAC addresses into account by default. To take them into account, activate the locally administered MAC addresses on the switch by using the CLI. Connect to the IBM BNT 1/10 Gb Switch Module in bay #2 of the chassis. Figure 5-12 shows how the `virt vmrmisc lmac` command enables those private MAC addresses.

```
Router>enable
Enable privilege granted.

Router#conf t
Enter configuration commands, one per line.  End with Ctrl/Z.

Router(config)#virt vmrmisc lmac
Router(config)#^Z

Router#write
```

Figure 5-12 Enabling locally administered MAC addresses

Tip: Locally administered MAC addresses are standard MAC addresses. They can be compared to the 192.168 class C subnet that is used to create local IP addresses.

MAC addresses have the format AA:BB:CC:DD:EE:FF, which is seven words in hexadecimal format. To recognize a locally administered MAC address, you need to convert the first word AA into binary. If the seventh bit is a 1, the MAC address is local. If it is a 0, it is universal. Universally administered MAC addresses are typically those created by the network vendors such as Broadcom and Intel.

5.4.3 Creating VM Groups

The scenario has a production environment where VLAN #720 and #721 exist. To use VMready functions such as implementing access control lists, associate the LPARs to VM Groups.

VM Groups are containers that substitute VLAN entities: a VM Group encapsulates a VLAN entity. VM Groups and VLANs use the same table of IDs, so they cannot coexist. To implement one VM Group with a specific VLAN ID, that VLAN ID must be removed from the VLAN table.

As a consequence, you need to remove VLANs #720 and #721 from the switch configuration. Then you need to create VM Groups #1 and #2 that are associated to VLANs #720 and #721. Finally, associate the LPARs to these VM Groups.

Remember: VLANs that are supposed to be used with LPARs are not defined in the usual VLAN table. Instead, they are defined in the VM Group table when VMready is enabled. The VLAN IDs and the VM Group VLAN IDs are common. You cannot have a VLAN and a VM group that share a VLAN ID.

When you create a VM Group, it automatically creates a VLAN ID in the VLAN table by default. This VLAN has the same ID as the VM Group VLAN ID, and is a reference to the new VM Group. You cannot modify the VLAN properties of a VM Group from the VLAN table. You must use the VM Group properties.

Attention: Make sure that you have no activity on the servers that are members of the VLAN that you are about to remove. Otherwise the servers will lose their connectivity. This is especially important in a production environment.

Creating VM Groups from the web interface

Before creating VM Groups, remove the current VLAN configuration. Open a web browser to the IBM BNT 1/10 Gb switch module. Click the Configuration tab, then click **IBM Networking OS 1/10Gb Uplink Ethernet Switch Module** → **Layer 2** → **Virtual LANs**, as shown in Figure 5-13. Select VLAN #720 and delete it. Then select VLAN #721 and delete it as well.

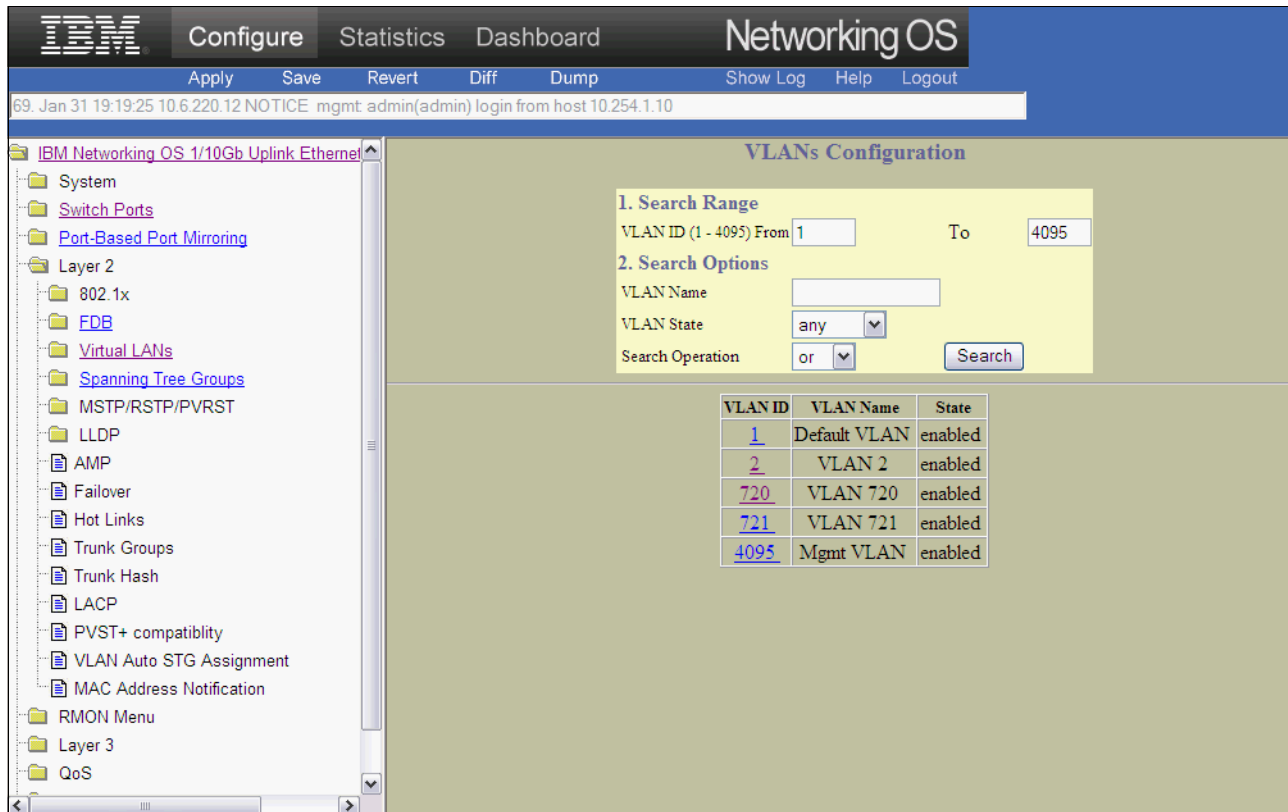


Figure 5-13 VLANs configuration

The next step is to create the VM Groups. To do so, click **IBM Networking OS 1/10Gb Uplink Ethernet Switch Module** → **Virtualization** → **Virtual Machines** → **VM Group General**. Click VM Group #1, which is not defined yet. Assign VLAN #720 to this entity. Enable Tagging and add EXT7 to the list of port to add to this group, as shown in Figure 5-14. Repeat the same action for VLAN #721.

Tip: Physical servers running bare metal operating systems without virtualization that used to be a member of the VLAN you removed need extra steps. You need to assign their ports back into the VM Group. To do so, use the **Port add** field in the VM Group configuration page and add your internal or external ports here.

Do not add the ports of your hypervisors that support the virtual machines in the VM Groups port configuration. VMready automatically configures the VLANs on the ports that support the partition network.

Remember: You cannot currently reach the virtual machines from the client. The physical ports on the switch are not aware of VLAN #720 and VLAN #721 yet. So all the traffic that comes from or to the partitions is blocked.

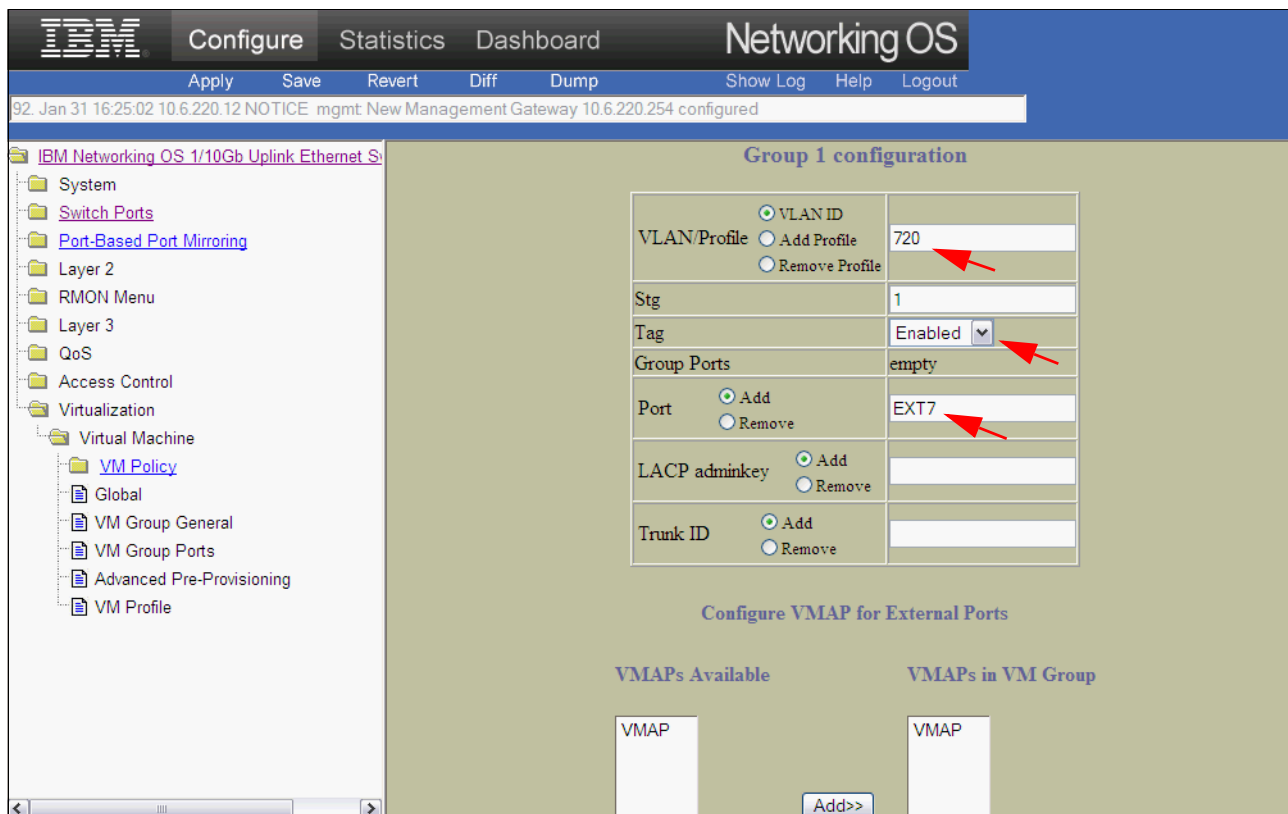


Figure 5-14 Creation of VMgroup #1

After the creation of the VM Groups, the partitions need to be added to these groups. You need the MAC addresses of your partitions to associate them to the groups that you created. Click the Configuration tab, then **IBM Networking OS 1/10Gb Uplink Ethernet Switch Module** → **Virtualization** → **Virtual Machines** → **VM Group General**. As shown in Figure 5-15, enter each MAC address and associate them to their related VM Groups:

- ▶ Web server is associated with VM Group #1 (VLAN #720)
- ▶ DB server is associated with VM Group #1 (VLAN #720)
- ▶ File server is associated with VM Group #2 (VLAN #721)

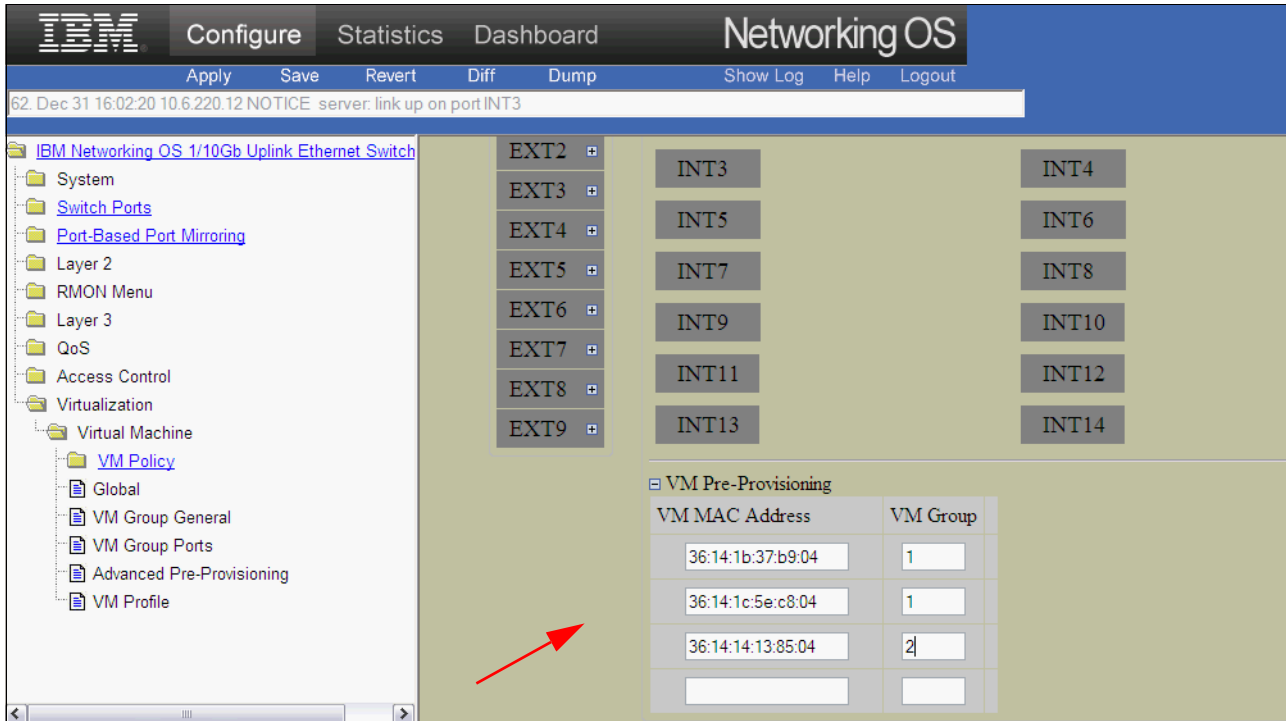


Figure 5-15 Pre-provisioning of the MAC addresses

After this step, you can generate traffic (like a ping) between your client workstation and the LPARs. Figure 5-16 shows that the LPARs were discovered on ports INT7 and INT8.

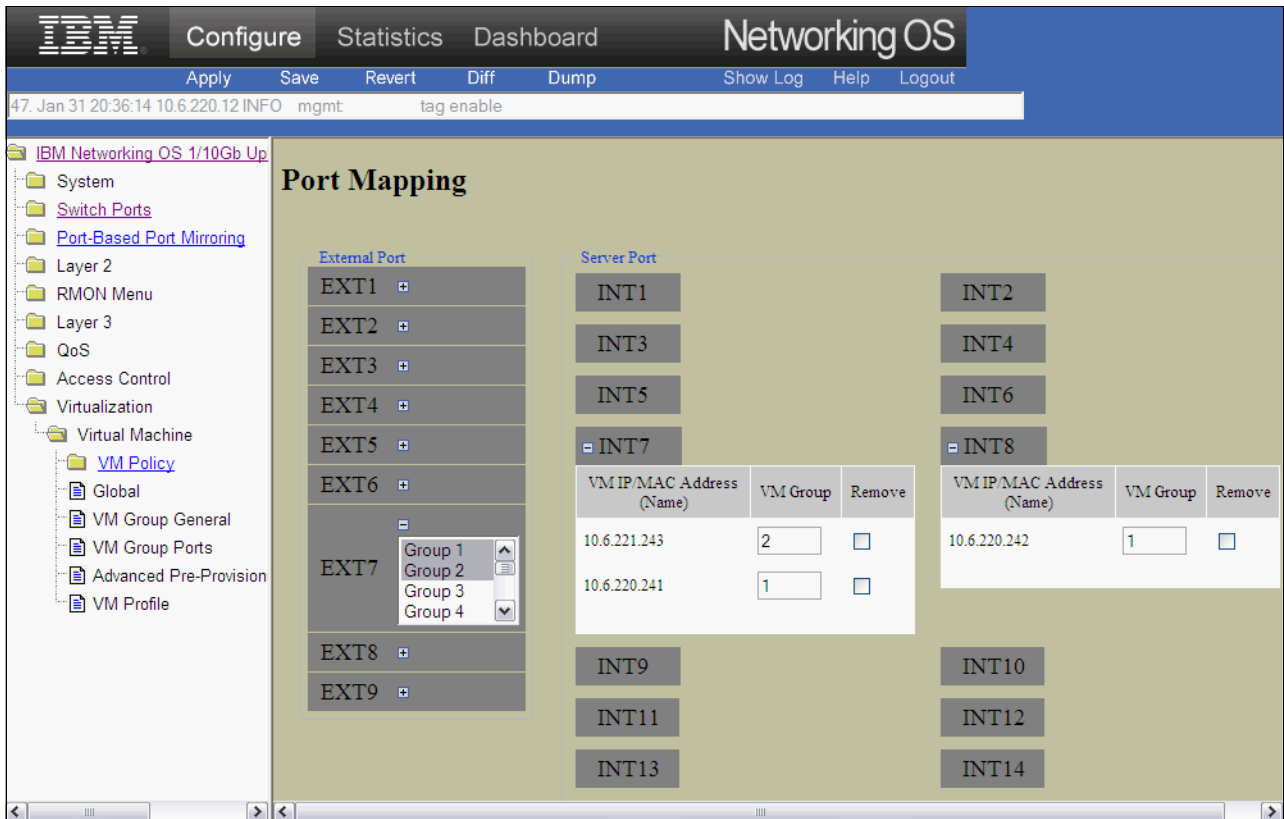


Figure 5-16 VM Group Ports overview

In the dashboard view of the Virtual LANs, you can see that port INT7 was automatically defined as member of VLAN #720 and #721 (Figure 5-17). Port INT8 is only member of VLAN #720 because no LPAR on VLAN #721 has been moved to that port yet.

The screenshot shows the 'VLANs Dashboard' in the IBM Networking OS interface. It includes a search range (1-4095) and search options. Below is a table of VLAN configurations:

VLAN ID	VLAN Name	VLAN Ports	VLAN Type	Private VLAN Type	Private VLAN Map	Private VLAN State	Management VLAN State	State
1	Default VLAN	INT1-INT6 INT9-INT14 EXT1-EXT9	Port based	empty		disabled	disabled	enable
2	VLAN 2	INT7 INT8 EXT7	Port based	empty		disabled	disabled	enable
720	VM Group 1 (T)	INT7 INT8 EXT7	Port based	empty		disabled	disabled	enable
721	VM Group 2 (T)	INT7 EXT7	Port based	empty		disabled	disabled	enable
4095	Mgmt VLAN	INT1-INT6 INT9 INT11-INT14 MGT1 MGT2	Port based	empty		disabled	enabled	enable

Figure 5-17 VLANs overview

Creating VM Groups from the CLI

This section shows how to perform these steps:

1. Remove the existing VLANs
2. Create the VM Groups and associate the LPARs to these groups

Figure 5-18 shows how to remove the current VLANs #720 and #721 from the configuration.

```
Router>show vlan
VLAN          Name                Status MGT          Ports
-----
1             Default VLAN       ena    dis    INT1-INT14 EXT1-EXT3
              EXT5-EXT9
2             VLAN 2             ena    dis    INT7 INT8 EXT7
720 VLAN 720       ena  dis  INT7 INT8 EXT7
721 VLAN 721       ena  dis  INT7 INT8 EXT7
4095         Mgmt VLAN         ena    ena    INT1-INT14 MGT1 MGT2

Router>enable
Enable privilege granted.

Router#conf t
Enter configuration commands, one per line.  End with Ctrl/Z.

Router(config)#no vlan 720

Router(config)#no vlan 721

Router(config)#^Z
Router#write
```

Figure 5-18 Removing the current VLANs

Figure 5-19 on page 123 shows how `virt vmgroup` command creates VM Groups #1 for VLAN #720 and VM Group #2 for VLAN #721. It also shows how it associates the LPARs into these groups. The `show virt vm` command displays the current view of the group assignments.

Tip: You need to generate network traffic before running the `show virt vm` command. Otherwise you will not see the same result.


```

Router>enable
Enable privilege granted.

Router#conf t
Enter configuration commands, one per line.  End with Ctrl/Z.

Router(config)#virt vmgroup 1 vlan 720
Router(config)#virt vmgroup 1 tag
Router(config)#virt vmgroup 2 vlan 721
Router(config)#virt vmgroup 2 tag
Router(config)#virt vmgroup 1 port EXT7
Router(config)#virt vmgroup 2 port EXT7
Router(config)#

Router(config)#virt vmgroup 1 vm 36:14:1b:37:b9:04
Router(config)#virt vmgroup 1 vm 36:14:1c:5e:c8:04
Router(config)#virt vmgroup 2 vm 36:14:14:13:85:04
Router(config)#^Z

Router#write

Router#sho virt vm
IP Address          VMAC Address          Index Port    VM Group (Profile)
-----
10.6.221.243        36:14:14:13:85:04    1    INT7    2
10.6.220.241        36:14:1b:37:b9:04    0    INT7    1
10.6.220.242        36:14:1c:5e:c8:04    2    INT8    1

Number of entries: 3

```

Figure 5-19 Creating the VM Groups and associating the LPARs

5.4.4 Applying access control lists

In the scenario, you want to block all the traffic of the non-secure connections that go to the Web server. To block them, create one VLAN Map (VMAP) to block all the packets that go to the Web server on port 80. A VMAP is like an access control list (ACL) but it is applied to a VM Group. The VMAP is manually associated to the VM Group that the VM belongs to.

This section addresses how to perform the following steps:

1. Creating one VMAP that blocks the traffic for Web server on port 80
2. Associating this VMAP to the VM Group

Applying ACL from the web interface

Connect to the IBM BNT 1/10G Switch Module in bay #2 with a web browser. Click the **Configuration** tab, then **IBM Networking OS 1/10Gb Uplink Ethernet Switch Module** → **Access Control** → **VLAN MAP** → **Add VMAP**. As shown in Figure 5-20, you can specify multiples restrictions on a particular partition or groups of partitions. In the example, block all the traffic that goes to the Web server on port 80. Set **Action** to **Deny**, enter the **Destination IP Address** of the Web server, and enable **Destination Port 80**.

Field	Value	Mask
VMAP Id (1 - 128)	1	
Group Id	0	
Filter Action	Deny	Set priority value: none
Ethernet Packet Format	Disabled	
Tagging Packet Format	Disabled	
IP Packet Format	None	
Source MAC Address	00:00:00:00:00:00	ff:ff:ff:ff:ff:ff
Destination MAC Address	00:00:00:00:00:00	ff:ff:ff:ff:ff:ff
Ethernet Type	None	Value (0600-fff): 600
VLAN Id (1-4095, 0 = any)	1	Mask (0-fff): fff
802.1p Priority	None	
Type of Service (0-255)	0	Disabled
Protocol (0-255)	0	Disabled
Source IP Address	0.0.0.0	255.255.255.255
Destination IP Address	10.6.220.241	255.255.255.255
TCP/UDP Src Port (1-65535)	1	Mask (1-fff): fff
TCP/UDP Dst Port (1-65535)	80	Mask (1-fff): fff
TCP Flags	<input type="checkbox"/> FIN <input type="checkbox"/> SYN <input type="checkbox"/> RST <input type="checkbox"/> PSH <input type="checkbox"/> ACK <input type="checkbox"/> URG	Mask(0-3f): 3f

Figure 5-20 VMAP configuration

After creating the VMAP, associate it to the VM Group that contains the web partition. To do so, click **IBM Networking OS 1/10Gb Uplink Ethernet Switch Module** → **Virtualization** → **Virtual Machine** → **VM Group General**. As shown in Figure 5-21, enter VM Group #1 to edit the configuration. Add your VMAP in the **Configure VMAPs for All Ports** list.

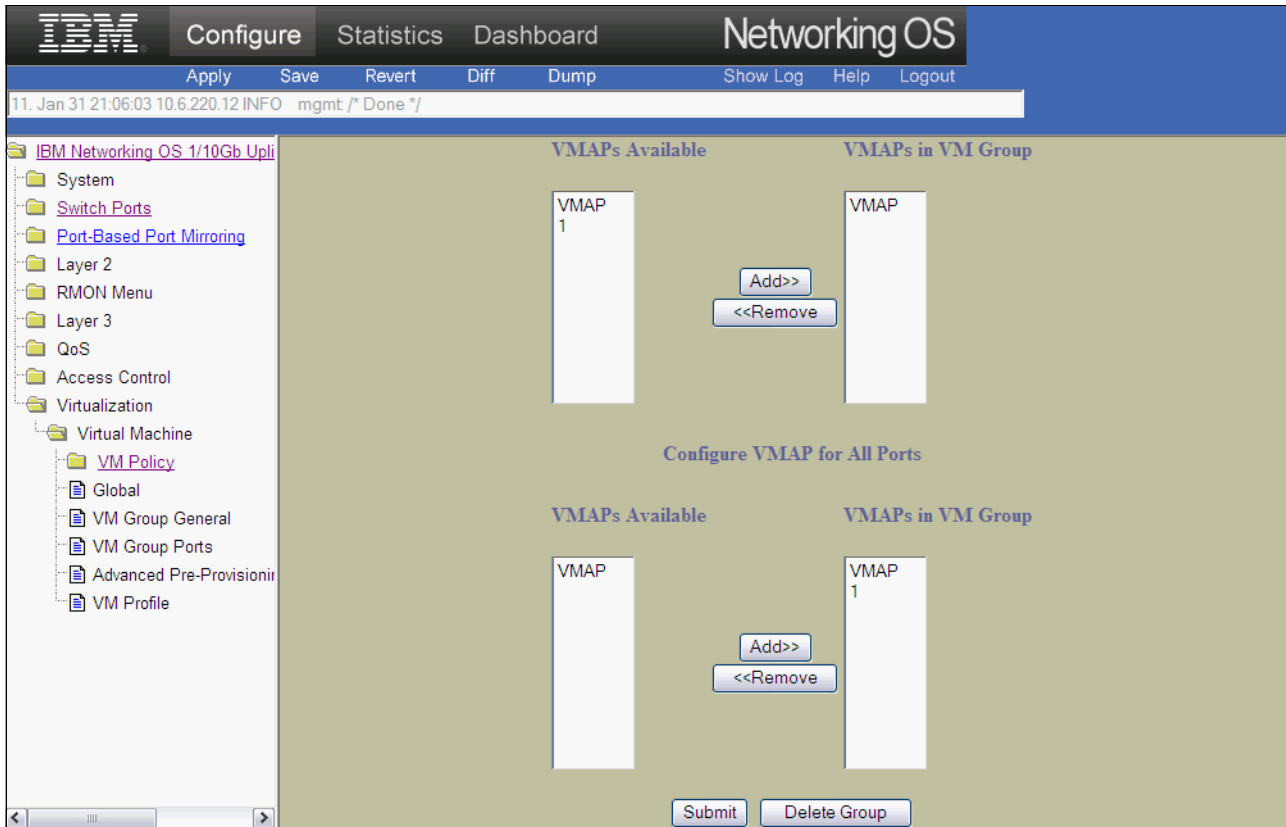


Figure 5-21 VMAP association within the VM Group

Applying ACL from the CLI

Connect to the IBM BNT 1/10 Gb Switch Module in bay #2 using a telnet session. As shown in Figure 5-22, the **access-control vmap** command creates VMAPs, and the **virt vmgroup** command associates those VMAPs to VM Groups.

```
Router>enable
Enable privilege granted.

Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.
Router(config)#

Router(config)#access-control vmap 1 action deny
Router(config)#access-control vmap 1 ipv4 destination-ip-address 10.6.220.241
Router(config)#access-control vmap 1 tcp-udp destination-port 80

Router(config)#virt vmgroup 1 vmap 1
Router(config)#^Z

Router#write
```

Figure 5-22 Applying VMAP with the CLI

5.4.5 Applying traffic shaping

The scenario restricts the bandwidth that is allocated to the File server. To do that, create a VM Policy and apply traffic shaping on the outgoing traffic of the LPAR.

Restriction: This policy applies only to traffic that goes through the physical switch. It does not apply between two LPARs on the same host whose traffic stays in the SEA.

Applying traffic shaping from the web interface

Connect to the IBM BNT 1/10 Gb Switch Module in bay #2 with a web browser. Click the Configuration tab, then **IBM Networking OS 1/10Gb Uplink Ethernet Switch Module** → **Virtualization** → **Virtual Machine** → **VM Policy** → **Add VM Policy**. Enter the MAC address

of File server, activate the Bandwidth Control, and set a maximum rate of 1 MB/s for outgoing traffic (Figure 5-23). Finally, select an ACL ID which is not used, #1 in the example. The policy takes effect immediately.



Figure 5-23 Traffic shaping configuration

Applying traffic shaping from the CLI

Figure 5-24 shows how `virt vmpolicy vmbwidth` command creates a VM Policy for traffic shaping. In the example, limit the outgoing traffic to 1 Mbit/s. The ACL ID which is created is #1.

```
Router>enable
Enable privilege granted.
Router#
Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.

Router(config)#virt vmpolicy vmbwidth 36:14:14:13:85:04 txrate 1024 1024 1
Router(config)#virt vmpolicy vmbwidth 36:14:14:13:85:04 bwctrl
Router(config)#^Z

Router#write
```

Figure 5-24 Applying traffic shaping

5.5 Validating the network policies

Now that all the network policies are enabled, you must validate that they are applied. This section describes these steps:

1. Validating the ACL policy
2. Validating the traffic shaping policy

5.5.1 Validating the ACL policy

To check that the VMAP is correctly configured, try to open a non-secure connection on Web server. Also, try to open a secure connection to make sure all the traffic is not blocked by the switch. Figure 5-25 shows that the traffic is blocked on port 80 of Web server.

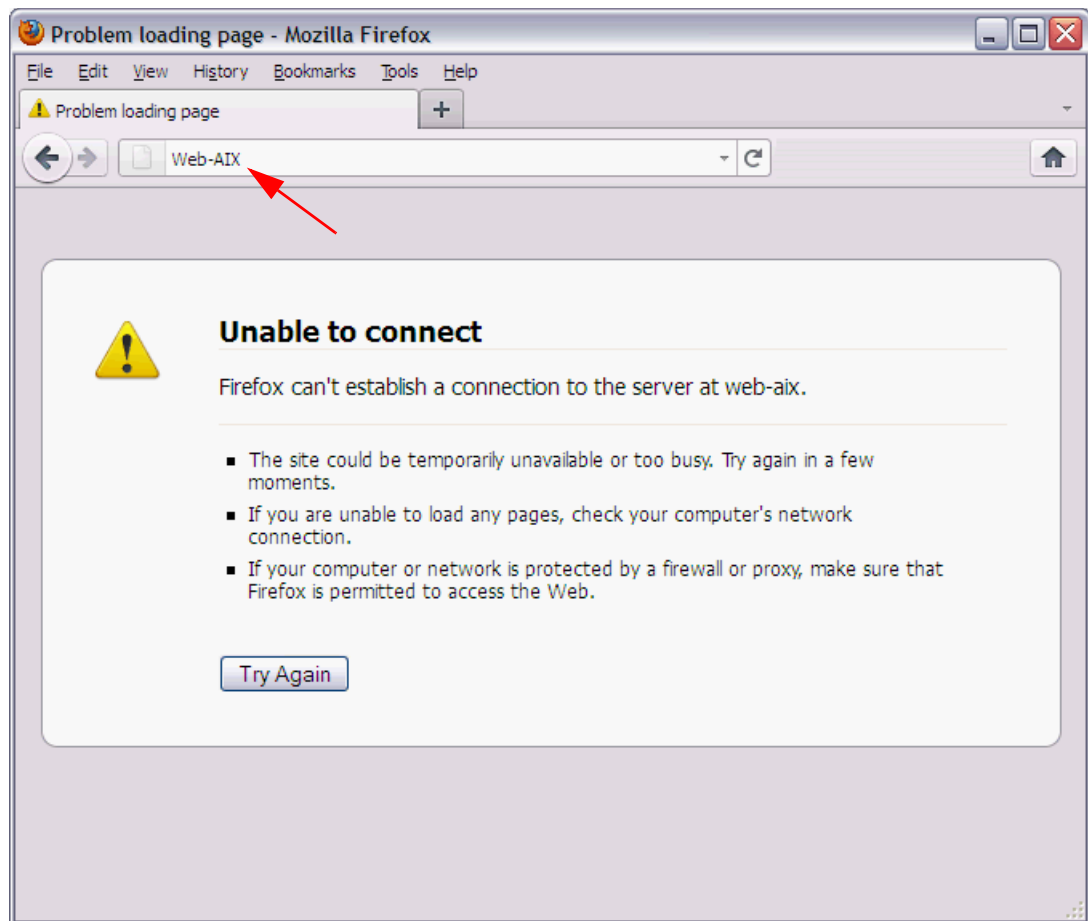


Figure 5-25 Home page of Web server with a non-secure connection on port 80

Figure 5-26 shows that Web server still responds to ping, but network traffic on port 80 is automatically blocked by the switch.

```
Administrator: Windows PowerShell
PS C:\Documents and Settings\Administrator> ping Web-AIX

Pinging Web-AIX [10.6.220.241] with 32 bytes of data:

Reply from 10.6.220.241: bytes=32 time=1ms TTL=251
Reply from 10.6.220.241: bytes=32 time=1ms TTL=251
Reply from 10.6.220.241: bytes=32 time=1ms TTL=251
Reply from 10.6.220.241: bytes=32 time=1ms TTL=251

Ping statistics for 10.6.220.241:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
PS C:\Documents and Settings\Administrator> telnet Web-AIX 80
Connecting To Web-AIX...Could not open connection to the host, on port 80: Connect failed
PS C:\Documents and Settings\Administrator> _
```

Figure 5-26 Test of the non-secure connection on port 80 of Web server

Figure 5-27 confirms that secure traffic is still active.



Figure 5-27 Home page of Web-AIX server with a secure connection on port 443

5.5.2 Validating the traffic shaping policy

To validate the network policy based on traffic shaping between the client workstation and File server, run the JPerf test again. Figure 5-28 shows that the network is now limited to 1 MBits/s.

```
# iperf -c 10.6.221.235 -P 1 -i 1 -p 5001 -f m -t 10 -T 1
-----
Client connecting to 10.6.221.235, TCP port 5001
TCP window size: 0.25 MByte (default)
-----
[ 3] local 10.6.220.243 port 61320 connected with 10.6.221.235 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec  0.25 MBytes 2.10 Mbits/sec
[ 3] 1.0- 2.0 sec  0.12 MBytes 1.05 Mbits/sec
[ 3] 2.0- 3.0 sec  0.12 MBytes 1.05 Mbits/sec
[ 3] 3.0- 4.0 sec  0.00 MBytes 0.00 Mbits/sec
[ 3] 4.0- 5.0 sec  0.12 MBytes 1.05 Mbits/sec
[ 3] 5.0- 6.0 sec  0.12 MBytes 1.05 Mbits/sec
[ 3] 6.0- 7.0 sec  0.00 MBytes 0.00 Mbits/sec
[ 3] 7.0- 8.0 sec  0.12 MBytes 1.05 Mbits/sec
[ 3] 8.0- 9.0 sec  0.12 MBytes 1.05 Mbits/sec
[ 3] 9.0-10.0 sec  0.00 MBytes 0.00 Mbits/sec
[ 3] 0.0-10.0 sec  1.09 MBytes 0.91 Mbits/sec
```

Figure 5-28 IPERF test

5.6 Validating Nmotion

After showing that the network policies are applied, validate that they are still active after a partition migration from one host to another.

During the initial tests, Web and File servers were hosted on PowerVM1 (port INT7 on the switch). The DB server was hosted on PowerVM2 (port INT8 on the switch). As shown in Figure 5-29, port INT7 was automatically configured to allow VLAN #720 and #721. Port INT8 was automatically configured to allow VLAN #720.

```

Router#sho vlan
VLAN          Name                Status MGT          Ports
-----
1             Default VLAN        ena   dis   INT1-INT6 INT9-INT14
              EXT1-EXT9
2             VLAN 2              ena   dis   INT7 INT8 EXT7
720         VM Group 1 (T)    ena dis INT7 INT8 EXT7
721         VM Group 2 (T)    ena dis INT7 EXT7
4095         Mgmt VLAN          ena   ena   INT1-INT6 INT9 INT11-INT14
              MGT1 MGT2

Router#sho virt vm
IP Address      VMAC Address          Index Port    VM Group (Profile)
-----
10.6.221.243    36:14:14:13:85:04    2   INT7     2
10.6.220.241    36:14:1b:37:b9:04    0   INT7     1
10.6.220.242    36:14:1c:5e:c8:04    1   INT8     1

Number of entries: 3

```

Figure 5-29 VLAN configuration overview before partition migration

Migrate the Web and File LPARs from PowerVM1 to PowerVM2, as shown in Figure 5-30.

Figure 5-30 LPAR overview

In Figure 5-31, you can see that port INT8 is reconfigured so it can allow VLAN #721 traffic.

```

Router#
Jan 31 22:21:28 10.6.220.12 NOTICE vm: 10.6.220.241 moved from port INT7 to
port INT8

Jan 31 22:22:54 10.6.220.12 NOTICE vm: 10.6.221.243 moved from port INT7 to
port INT8

Router#sho vlan
VLAN          Name                Status MGT          Ports
-----
1      Default VLAN      ena    dis  INT1-INT6 INT9-INT14
                                EXT1-EXT9
2      VLAN 2            ena    dis  INT7 INT8 EXT7
720 VM Group 1 (T)      ena    dis  INT8 EXT7
721 VM Group 2 (T)      ena    dis  INT8 EXT7
4095  Mgmt VLAN          ena    ena  INT1-INT6 INT9 INT11-INT14
                                MGT1 MGT2

Router#sho virt vm
IP Address      VMAC Address        Index Port    VM Group (Profile)
-----
10.6.221.243    36:14:14:13:85:04   2    INT8    2
10.6.220.241    36:14:1b:37:b9:04   0    INT8    1
10.6.220.242    36:14:1c:5e:c8:04   1    INT8    1

Number of entries: 3
Router#

```

Figure 5-31 VLAN configuration overview after partition migration

5.6.1 Validating the ACL policy

Run the test again to prove that the network policy moved with Web server migration. As shown in Figure 5-32, port 80 is still blocked.

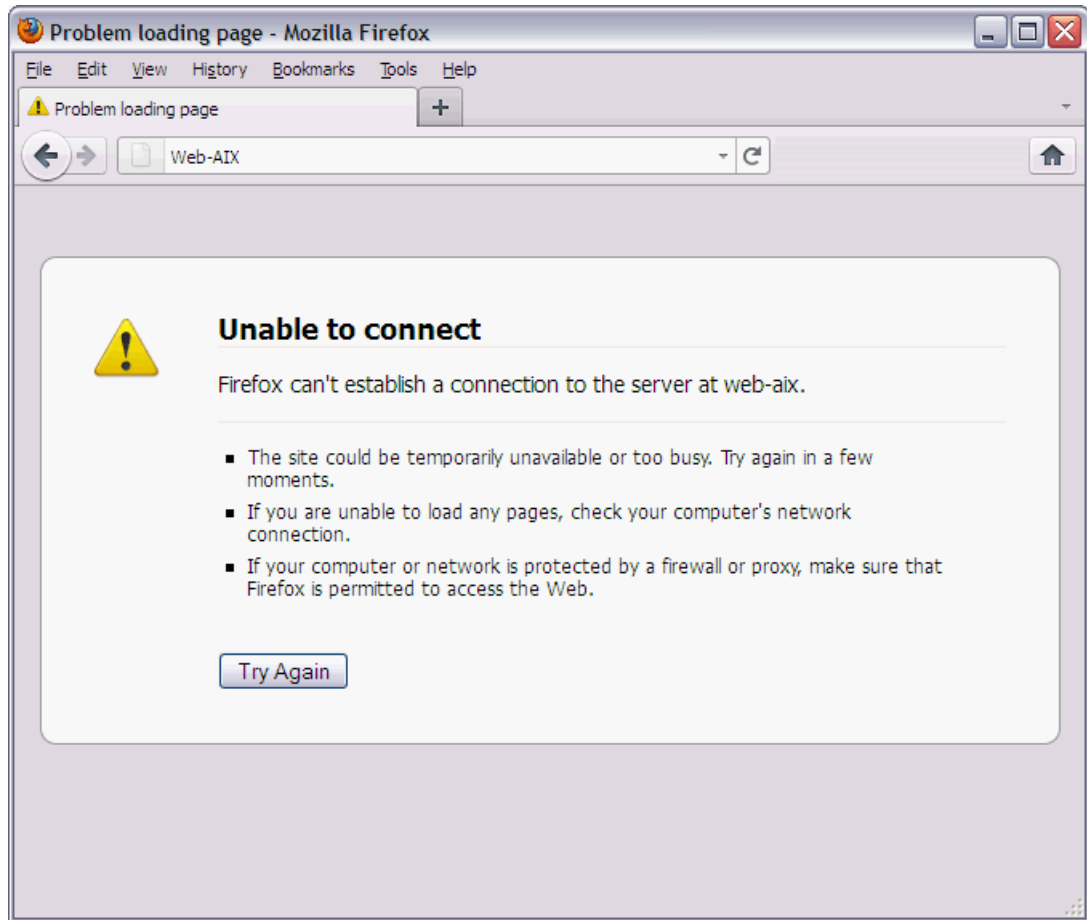


Figure 5-32 Non-secure connection on Web server

5.6.2 Validating the traffic shaping policy

Figure 5-33 shows the network traffic on File server before, during, and after the migration. At 10s, the partition migration is over. You can see that the policy is still active after the migration.

```
# iperf -c 10.6.221.235 -P 1 -i 1 -p 5001 -f m -t 10 -T 1
-----
Client connecting to 10.6.221.235, TCP port 5001
TCP window size: 0.25 MByte (default)
-----
[ 3] local 10.6.220.243 port 61534 connected with 10.6.221.235 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec  0.00 MBytes  0.00 Mbits/sec
[ 3] 1.0- 2.0 sec  0.25 MBytes  2.10 Mbits/sec
[ 3] 2.0- 3.0 sec  0.00 MBytes  0.00 Mbits/sec
[ 3] 3.0- 4.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 4.0- 5.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 5.0- 6.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 6.0- 7.0 sec  0.25 MBytes  2.10 Mbits/sec
[ 3] 7.0- 8.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 8.0- 9.0 sec  0.00 MBytes  0.00 Mbits/sec
[ 3] 9.0- 10.0 sec  0.00 MBytes  0.00 Mbits/sec
[ 3] 10.0- 11.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 11.0- 12.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 12.0- 13.0 sec  0.00 MBytes  0.00 Mbits/sec
[ 3] 13.0- 14.0 sec  0.25 MBytes  2.10 Mbits/sec
[ 3] 14.0- 15.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 15.0- 16.0 sec  0.25 MBytes  2.10 Mbits/sec
[ 3] 16.0- 17.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 17.0- 18.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 18.0- 19.0 sec  0.12 MBytes  1.05 Mbits/sec
[ 3] 19.0- 20.0 sec  0.12 MBytes  1.05 Mbits/sec
```

Figure 5-33 IPERF test



Implementing VMready to support KVM

This chapter describes the steps involved in implementing VMready for KVM environments. This chapter includes the following sections:

- ▶ Overview of VMready for KVM
- ▶ Implementation
- ▶ Implementing IEEE 802.1Qbg with VMready

6.1 Overview of VMready for KVM

IBM VMready allows you to configure virtual machine networking at the physical switch layer. This process provides a seamless interface for configuring both physical and virtual server networking. In a KVM environment, VMready ensures that network policies follow virtual machines as they migrate from one physical switch port to another using a technology called NMotion. This process makes the manual configuration of each individual switch ports to cater for virtual machine networking unnecessary.

After the VM-Aware network is configured by the network administrator, the VMready switch continues to ensure that connectivity and appropriate network policies are enforced. These policies are enforced for both virtual and physical servers.

6.1.1 A little about KVM networking

Guest networking in KVM can be implemented in different ways:

- ▶ Using Network Address Translation (NAT) where the host network interfaces acts as a router for the guests
- ▶ Using bridged networking where the host network interfaces forwards guest traffic to the physical switch
- ▶ Using directly attached physical devices that use PCI device assignment where the guest is given direct access to the hardware
- ▶ Using directly allocated virtual functions with PCIe SR-IOV where a physical network interface is carved into multiple virtual interfaces (also known as virtual functions)

Many implementations use bridged networking along with 802.1Q (VLAN tagging) to have a layer 2 segmentation between different groups of guests.

In a typical configuration, the physical switch ports to which a hypervisor is attached is configured in trunk mode, and is attached to multiple VLANs. In KVM, several virtual bridge interfaces are defined on the host that use different VLAN IDs.

In a network that is not virtual machine aware, network administrators must configure VLANs on physical switch ports. However, you have no further means of enforcing network policies such as access control lists (ACLs) or quality of service (QoS) policies on a per virtual machine basis.

In addition, all switch ports to which a given virtual machine could potentially move to (using live migration) need to be configured. This configuration includes the VLAN and network policies that the virtual machine belongs to.

As described in Chapter 2, “Introducing VMready” on page 11, VMready and 802.1Qbg help give control back to the network administrator. They remove the administrative burden of having to configure multiple physical switch ports.

For kernel-based virtual machine (KVM), VMready and 802.1Qbg have the following benefits:

- ▶ VMready enables per virtual machine network policies and automates physical switch port configuration
- ▶ 802.1Qbg increases the scalability of VMready and provides a more streamlined process to implement those network policies and assign them to virtual machines

6.1.2 VMready capabilities in a KVM environment

VMready provides enhanced granularity to physical switch network capabilities. It extends the capability of the switch from specifying network parameters at the physical port level down to the virtual machine level.

The following functions are available when configuring parameters per virtual machine port:

- ▶ VLAN membership
- ▶ Traffic shaping and monitoring
- ▶ ACLs
- ▶ QoS attributes

6.1.3 VMready terminology

Before describing how to configure VMready, couple of VMready specific terms need to be defined:

- ▶ **vmgroup:** This is a VM-Aware enhanced VLAN in which you can place virtual machines and associate vmaps. A VLAN ID can be directly assigned to a vmgroup. After a vmgroup is associated with a VLAN ID, it becomes the new point of configuration for that VLAN. Physical switch ports, static trunks, and LACP trunks can also be added to a vmgroup just like a regular VLAN.
- ▶ **vmap:** This is a specialized ACL that you can apply to vmgroups and VLANs.

6.1.4 Example setup

The implementation and example scenario is based on two host systems, three virtual machines (FILE, WEB, and DB), and a client system (Figure 6-1). The WEB and DB virtual machines host a two tier web application, and belong to the Finance department. The FILE virtual machine provides file sharing services for the Human Resources department.

The client system is used to demonstrate the impact of implementing network policies and to show that those policies that are preserved as virtual machines are migrated.

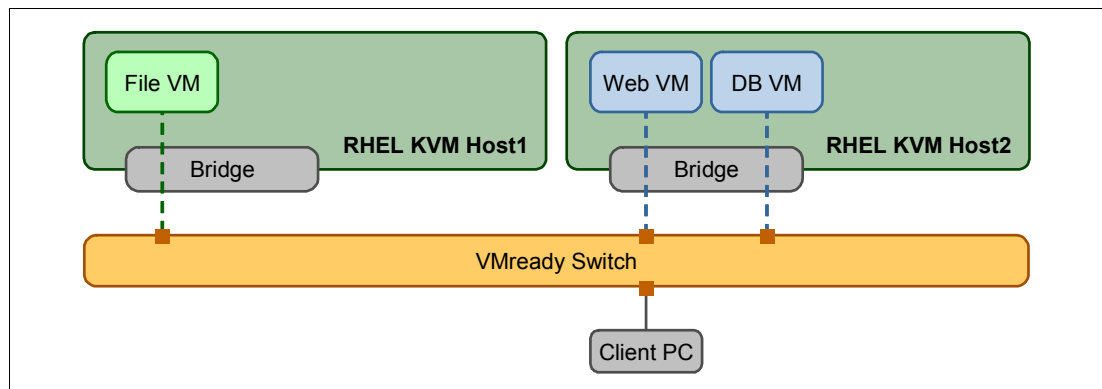


Figure 6-1 Lab logical layout

Figure 6-2 shows the physical layout of the implementation before VMready.

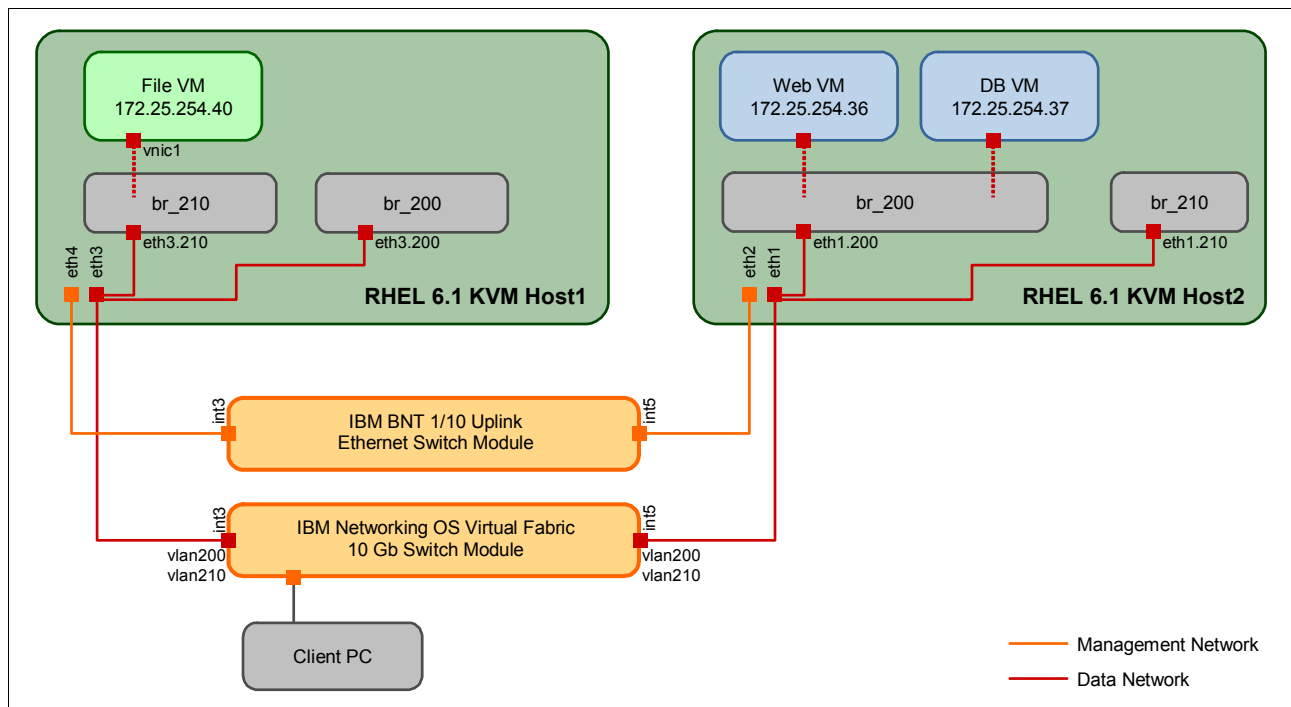


Figure 6-2 Lab physical layout

Each host has these connections:

- ▶ One 10 Gbps connection to an IBM Networking OS Virtual Fabric 10 Gb Switch Module running firmware version 6.8.0.66 for virtual machine networking
- ▶ One 1 Gbps connection to an IBM BNT 1/10 Uplink Ethernet Switch module for the management network

For the first host (called kvm1), both interfaces are connected to the INT3 port on each switch.

For the second host (called kvm2), both interfaces are connected to the INT5 port on each switch.

Each host is running RedHat Enterprise Linux 6.1 with the following virtualization packages:

- ▶ qemu-kvm-0.12.1.2-2.160.el6_1.8.x86_64
- ▶ libvirt-0.8.7-18.el6_1.1.x86_64
- ▶ virt-manager-0.8.6-4.el6.noarch

The FILE virtual machine starts in the same VLAN as the client PC (VLAN 210). The WEB and DB virtual machines start in a different VLAN (VLAN 200). Network traffic between VLAN200 and VLAN210 is not routed. The physical ports on the 10 Gbps physical switch are configured to belong to VLAN200 and VLAN210.

6.1.5 The example scenario

These steps will be described to demonstrate the VMready function:

1. Start the environment and see what it looks like without VMready
2. Enable VMready and show the discovery of the virtual machines

3. Implement network policies and demonstrate their impact
4. Migrate a virtual machine and show that its network settings are automatically moved with it.

6.2 Implementation

This section and the 802.1Qbg section use a scenario-based approach to illustrate the actual steps required to enable VMready in a KVM environment. This example is in a corporate IT environment.

The goal is to implement three basic corporate network policies:

1. Strict layer 2 network separation must be maintained between applications that belong to different business units.
2. For the Finance department, no web application server should be accessible on the standard http port (TCP 80)
3. Business units that host applications on shared IT infrastructure must be restricted in the amount of the shared resource that is made available to them. The intention of the policy is two fold:
 - To allow fair sharing of available resources in multi-tenant environments to prevent one department from monopolizing IT resources to the detriment of another
 - To ensure that departments are allocated resources in proportion to their contribution to IT budgets

To implement those policies, you must perform these steps:

1. Enable VMready on the physical switch
2. Create VM Groups that match the actual VLAN implementation and assign the virtual machines in the appropriate VM Group
3. Create a vmap to enforce policy number 2 (blocking http traffic on standard port) and assign that VMAP to the appropriate VM Group
4. Create a virtual machine policy to enforce policy number 3 and limit the network bandwidth available to the FILE virtual machine

6.2.1 Step 1: What the environment looks like without VMready

As shown in Figure 6-3, kvm1 and kvm2 are started, and the three virtual machines (WEB, DB, and FILE) are running.

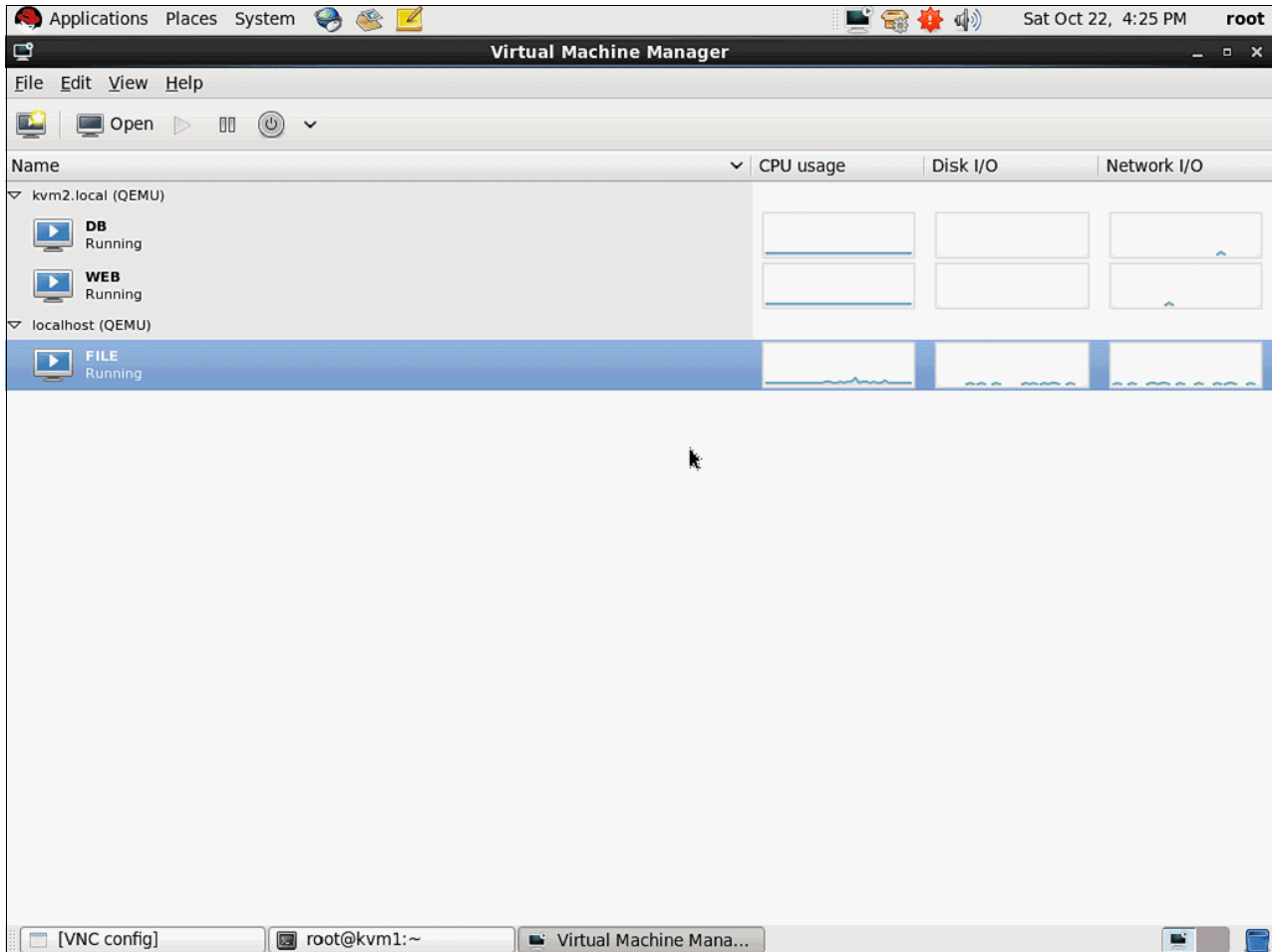


Figure 6-3 Virtual Machine Manager

Connect to the physical switch. Click the **Configure** tab, and then click **Virtualization** → **Virtual Machine** → **VM Group Ports**. No virtual machines are displayed on the internal ports, as shown in Figure 6-4.

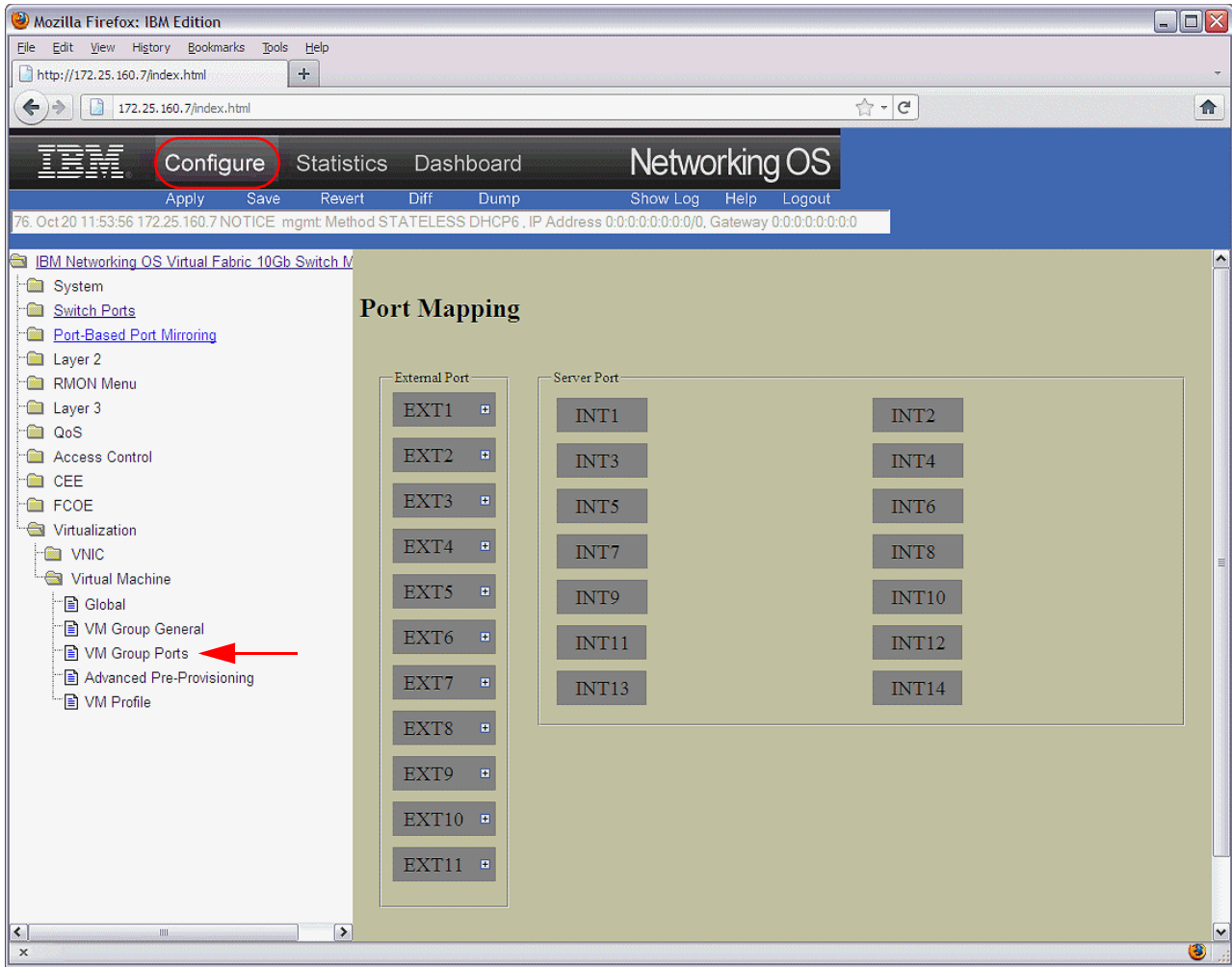


Figure 6-4 Without VMready

6.2.2 Step 2: Configuring VMready

To enable VMready for KVM, configure your physical switch to treat locally administered MAC addresses as virtual machines. This configuration is necessary with KVM because libvirt uses a 52:54:00 MAC addressing scheme by default, which is not a registered Organizationally Unique Identifier (OUI).

As explained in Chapter 2, “Introducing VMready” on page 11, VMready discovers virtual machines at the layer 2, based on their MAC address. By default, VMready switches are configured with the major hypervisor vendors OUI (such as VMware or Hyper-v). However, they are not configured to consider locally administered OUI as virtual machines.

Remember: Use `enable` and `configure terminal` mode when using the ISCLI to configure.

To enable the **Treat locally administered MAC addresses as virtual machines** feature, connect through telnet to the physical switch. Select the command-line interface (CLI), and enter one of the following commands:

- ▶ `/cfg/virt/vmrmisc/1macena` (ibmnos)
- ▶ `virt vmrmisc 1mac` (ISCLI)

Restriction: This command is only available in release 6.8 and later of the IBM Systems Networking OS. In previous versions, you cannot specify additional OUIs, and locally administered MAC addresses are not considered as virtual machines.

To apply the configuration, enter **apply** at the CLI prompt, then **save** to preserve the configuration at the next switch reboot. Enter **write** if you are using the ISCLI.

The rest of the configuration can be done from the physical switch web interface. However, corresponding CLI commands are included in both the IBM Networking OS (ibmnos) and the Industry Standard CLI (ISCLI).

Enable VMready by clicking **Virtualization** → **Virtual Machines** → **Global**.

As shown in Figure 6-5, select **enabled** from the **Virtual Machine Groups** list, then click **Submit**. Make sure that you click **Apply** and **Save** in the blue toolbar to apply the current configuration. You must save the changes so they are still active after the next physical switch reboot.

Remember: Always select **Apply** every time you make a configuration change in the physical switch and want that change to take effect. If you also want that change to remain after a physical switch reboot, also select **Save**. When entering CLI commands, always enter **apply** and **save** when you are using the **ibmnos**, or **write** when using the **ISCLI**.

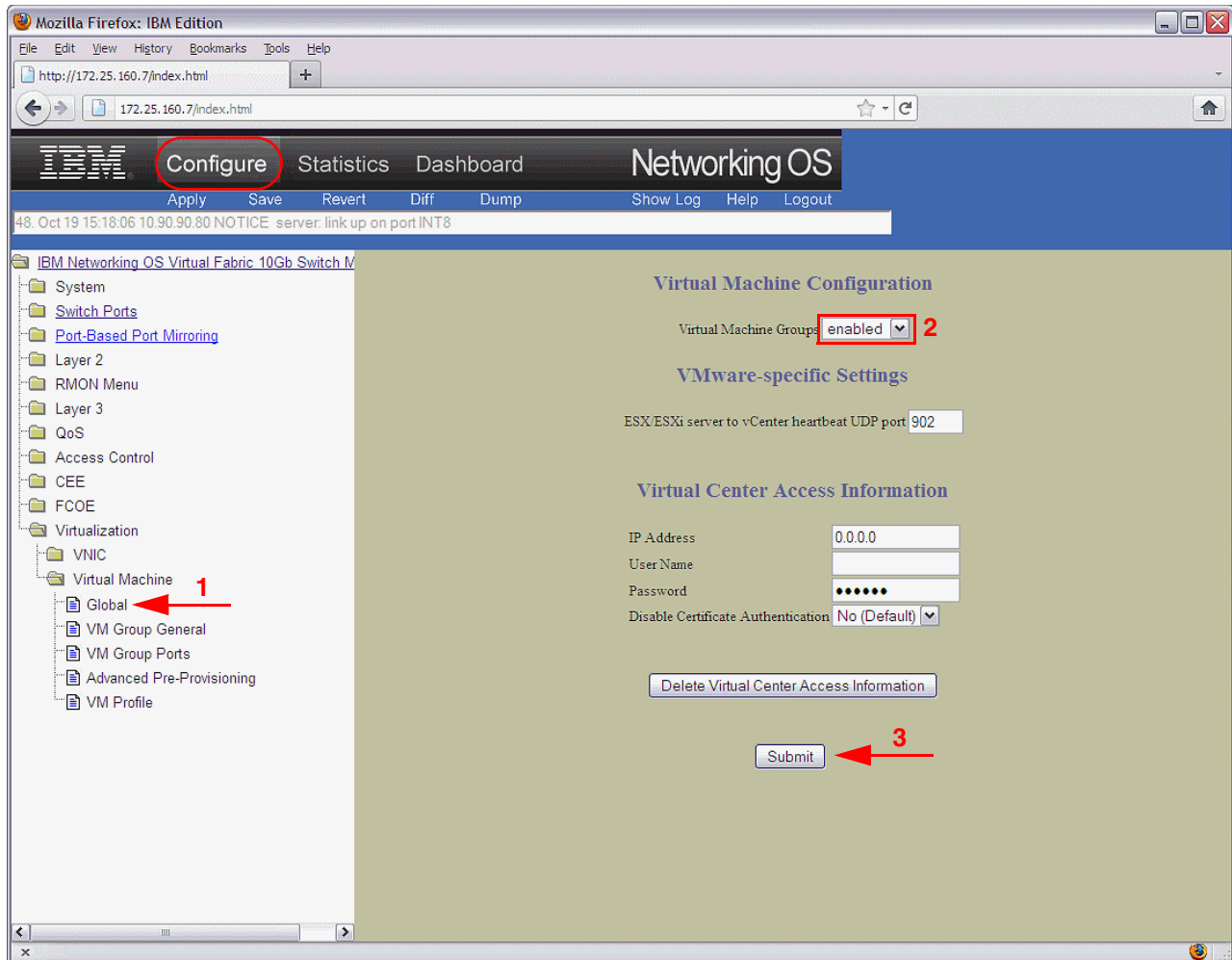


Figure 6-5 Enabling VMready

To do the same from the **ibmnos** CLI, enter the `/cfg/virt/enavmr` command. If you prefer to use the **ISCLI**, enter `virt enable`.

Click **Virtualization** → **Virtual Machine** → **VM Group Ports** view to see that the three virtual machines are discovered on the internal ports (Figure 6-6).

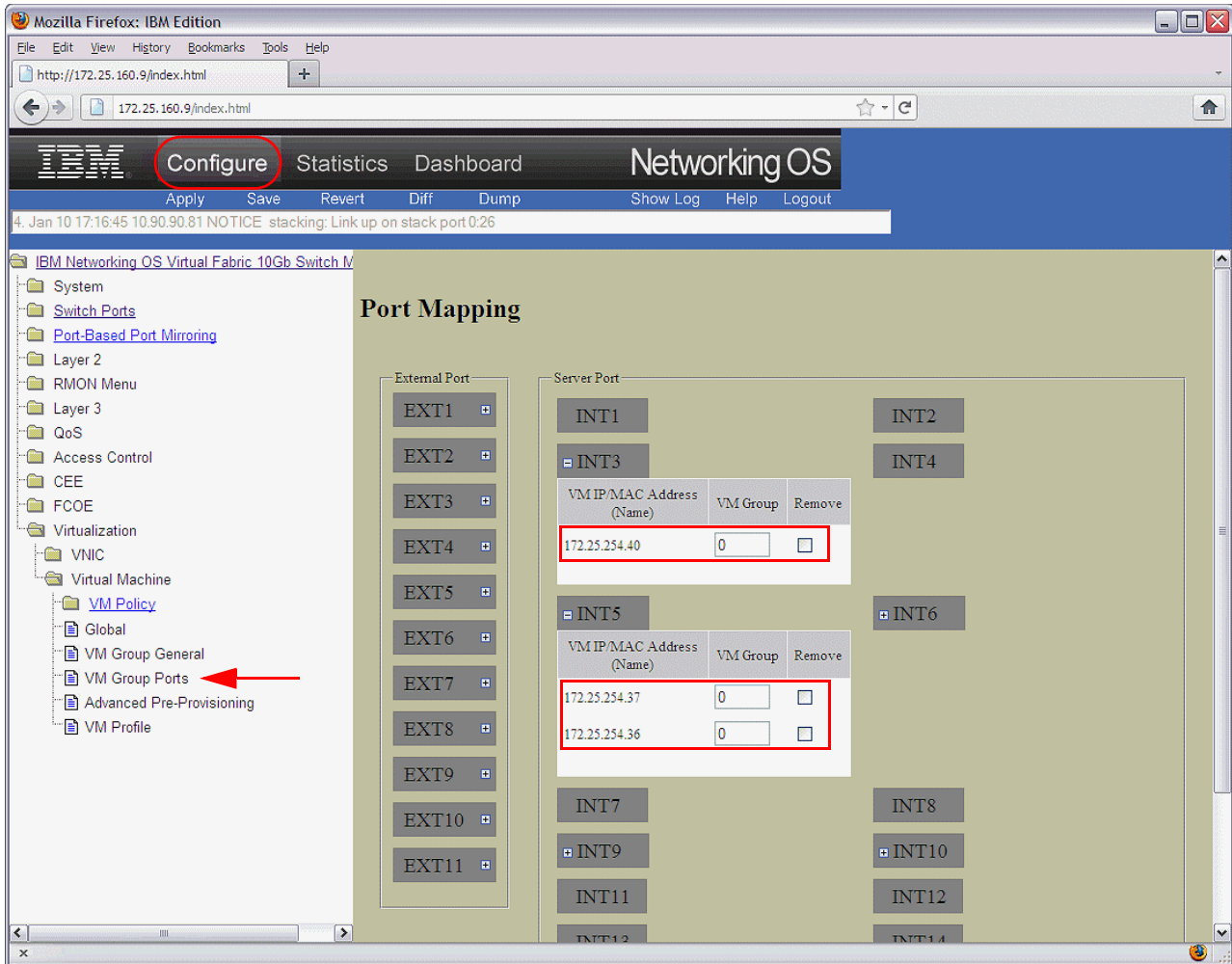


Figure 6-6 Viewing virtual machines with VMready enabled

To view virtual machines on internal ports by using the CLI, enter one of the following commands:

- ▶ `/info/virt/vm/dump` (ibmnos)
- ▶ `show virt vm` (ISCLI)

Remember: VMready discovers virtual machines at the layer 2, so some network traffic needs to be generated before they are displayed on **VM Group Ports** window. If layer 3 traffic is generated, VMready displays the virtual machines using their IP address.

You are now ready to apply network configuration to your virtual machines.

6.2.3 Step 3: Creating a VM Group and assigning virtual machines to it

Network policies are applied to virtual machines by using VM groups. VM groups have a VLAN ID assigned to them, which must be unique. This configuration means that you cannot create a VM Group with VLAN ID 200 if VLAN 200 exists on your switch. This is the case in the current setup as shown in Example 6-1.

Example 6-1 Existing VLAN configuration in the lab physical switch

```
>> Main# /info/12/vlan
VLAN          Name                Status MGT          Ports
-----
1      Default VLAN      ena   dis   INT1-INT14 EXT1-EXT11
200    VLAN 200          ena   dis   INT3 INT5
210    VLAN 210          ena   dis   INT3 INT5 INT6
4095   Mgmt VLAN        ena   ena   INT1-INT14 MGT1 MGT2
```

Note that INT6 is the physical switch port where the client PC is located.

In the scenario, because VLAN 200 and 210 already exist, you need to delete those VLANs as shown in Example 6-2.

Example 6-2 Deleting VLAN 200 and VLAN 210

```
/12/vlan 200/del
About to delete VLAN 200 - "VLAN 200"
Confirm deleting of this VLAN [y/n]: y
VLAN 200 with name "VLAN 200" is deleted.

>> Layer 2# /cfg/12/vlan 210/del
About to delete VLAN 210 - "VLAN 210"
Confirm deleting of this VLAN [y/n]: y
VLAN 210 with name "VLAN 210" is deleted.
```

Attention: When you delete those VLANs, all virtual machines and physical servers on that switch or stack of switches lose network connectivity. Therefore, you need a network outage on those VLANs when migrating to VMready if you already have VLANs implemented.

To create VM groups, click the **Configure** tab, then click **Virtualization** → **Virtual Machine** → **VM Group General**. Then click one of the VM groups in the table as shown in Figure 6-7.

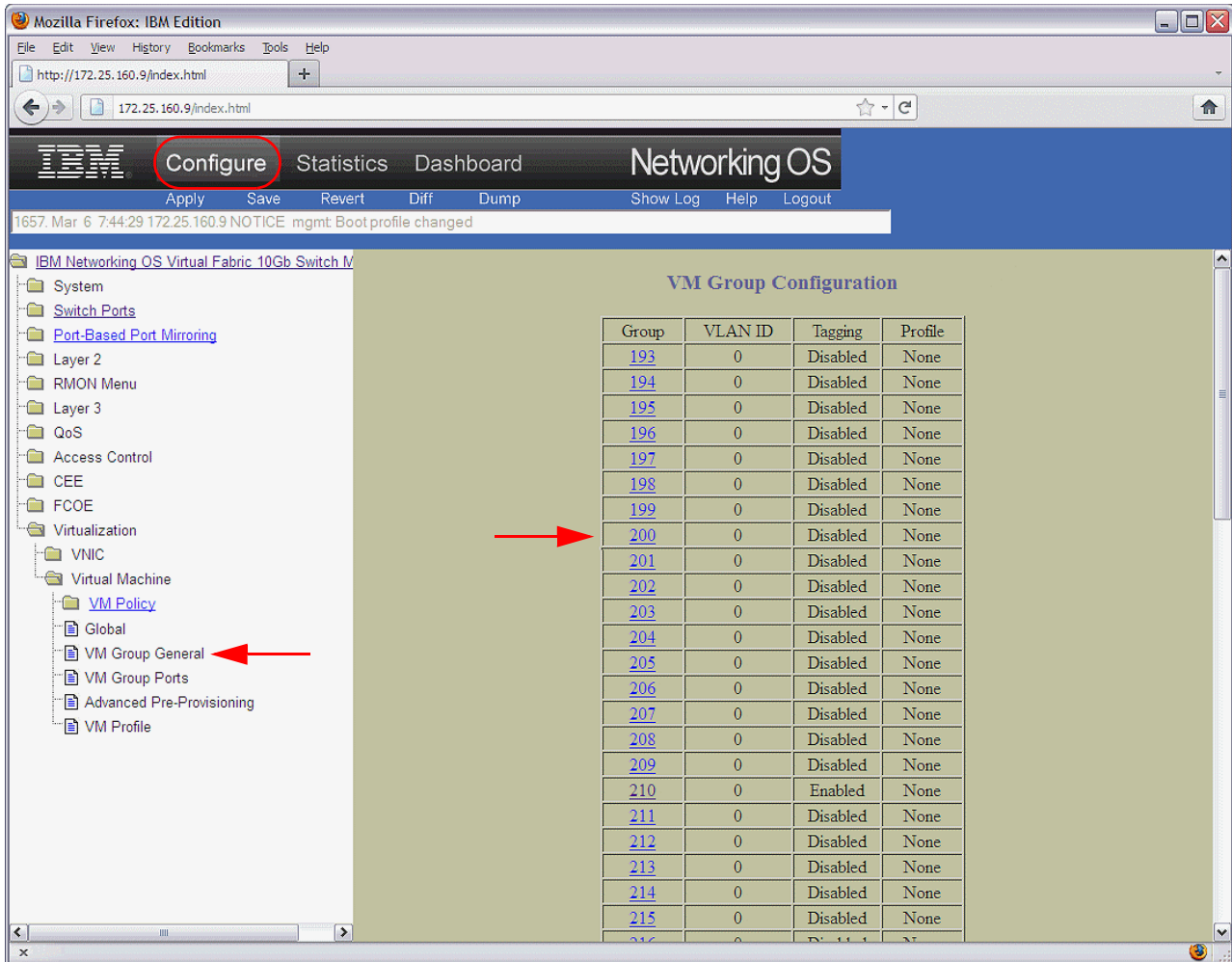


Figure 6-7 Configuring a VM group (Step 1)

A new window opens. Enter a VLAN ID for your VM group, enable VLAN tagging, and then scroll down to the bottom of the page and click **Submit**. Then click **Apply** and **Save** as shown in Figure 6-8.

Requirement: If you do not enable VLAN tagging on your VM Group, your virtual machines will not be able to receive network traffic from servers that do not belong to the same VM Group. This restriction includes a physical server somewhere else in the network.

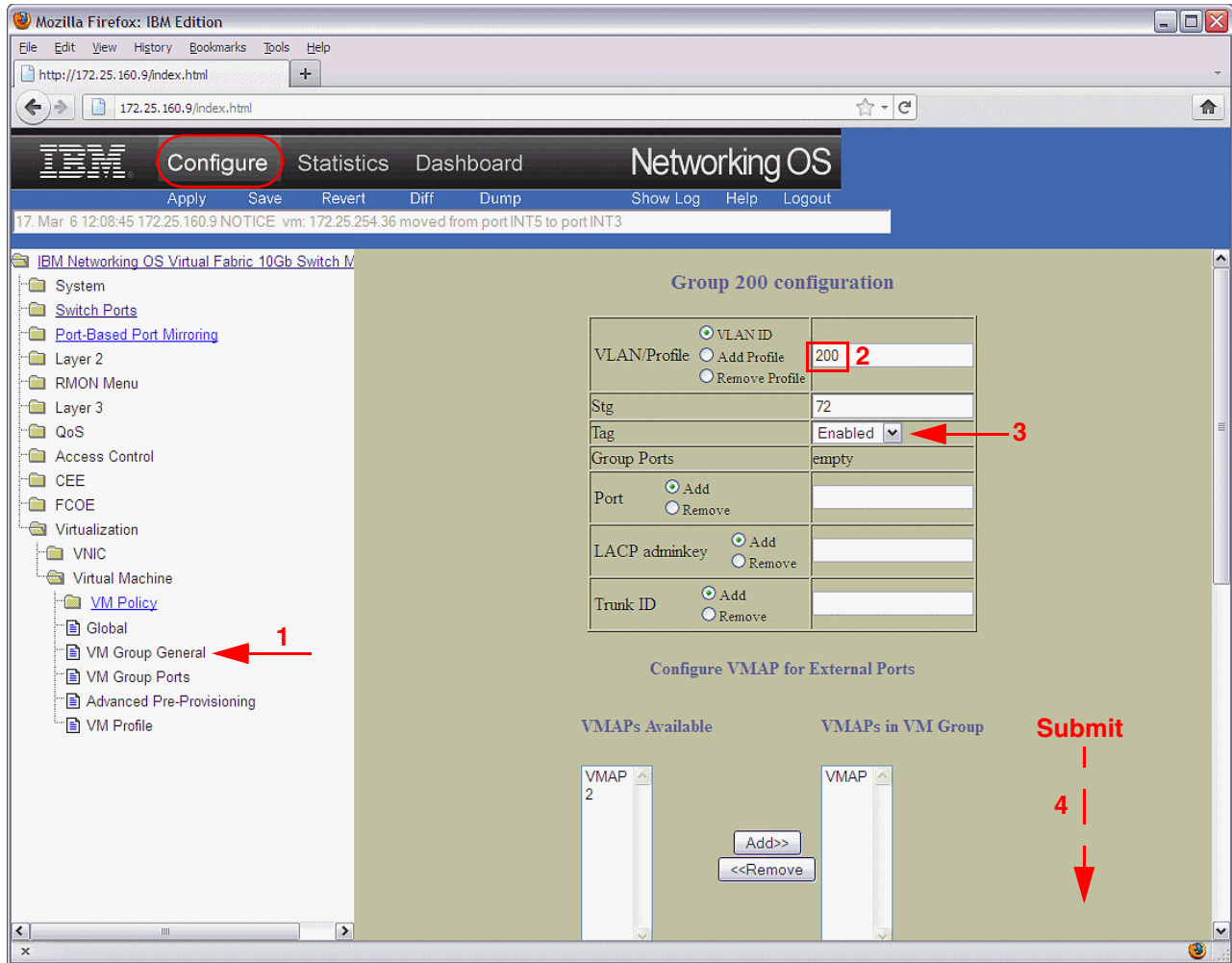


Figure 6-8 Configuring a VM group (Step 2)

Example 6-3 shows how to create a VM Group by using the ibmnos CLI.

Example 6-3 Creating a VM Group by using the ibmnos CLI

```

/cfg/virt/vmgroup/
Enter group number: (1-1024) 210
-----
[VM group 210 Menu]
vlan      - Set the group's vlan (only for groups with no VM profile)
vmap     - Set VMAP for this group
tag      - Enable vlan tagging on all VM group ports
    
```

```
addvm    - Add a virtual entity to the group
remvm    - Remove a virtual entity from the group
addprof  - Add a VM profile to the group
remprof  - Delete any VM profile associated with the group
addport  - Add ports to the group
remport  - Remove ports from the group
addtrunk - Add trunk to the group
remtrunk - Remove trunk from the group
addkey   - Add LACP trunk to the group
remkey   - Remove LACP trunk from the group
stg      - Assign VM group vlan to a Spanning Tree Group
del      - Delete group
cur      - Display current group configuration
```

```
>> VM group 210# vlan 210
```

```
>> VM group 210# apply
```

```
VLAN 210 was assigned to STG 72.
```

```
-----  
Apply complete; don't forget to "save" updated configuration.
```

```
>> VM group 210# save
```

```
Request will first copy the FLASH "active" config to "backup",  
then overlay FLASH "active" with new config.
```

```
Confirm saving to FLASH [y/n]: y
```

```
New config successfully saved to FLASH.
```

The equivalent in ISCLI is **virt vmgroup 210 vlan 210**.

You are now ready to assign virtual machines to your VM groups. Assign the WEB and DB virtual machines to the VM group 200 and the FILE virtual machine to the VM group 210. Then associate their MAC addresses to the VM groups.

You can also use the web graphical user interface (GUI, also called the BBI or browser-based interface). Click the **Configure** tab, then click **Virtualization** → **Virtual Machine** → **VM Group Ports**. Click the “+” sign next to the **VM Pre-provisioning** menu and enter the virtual machines MAC addresses along with their corresponding VM group as shown in Figure 6-9.

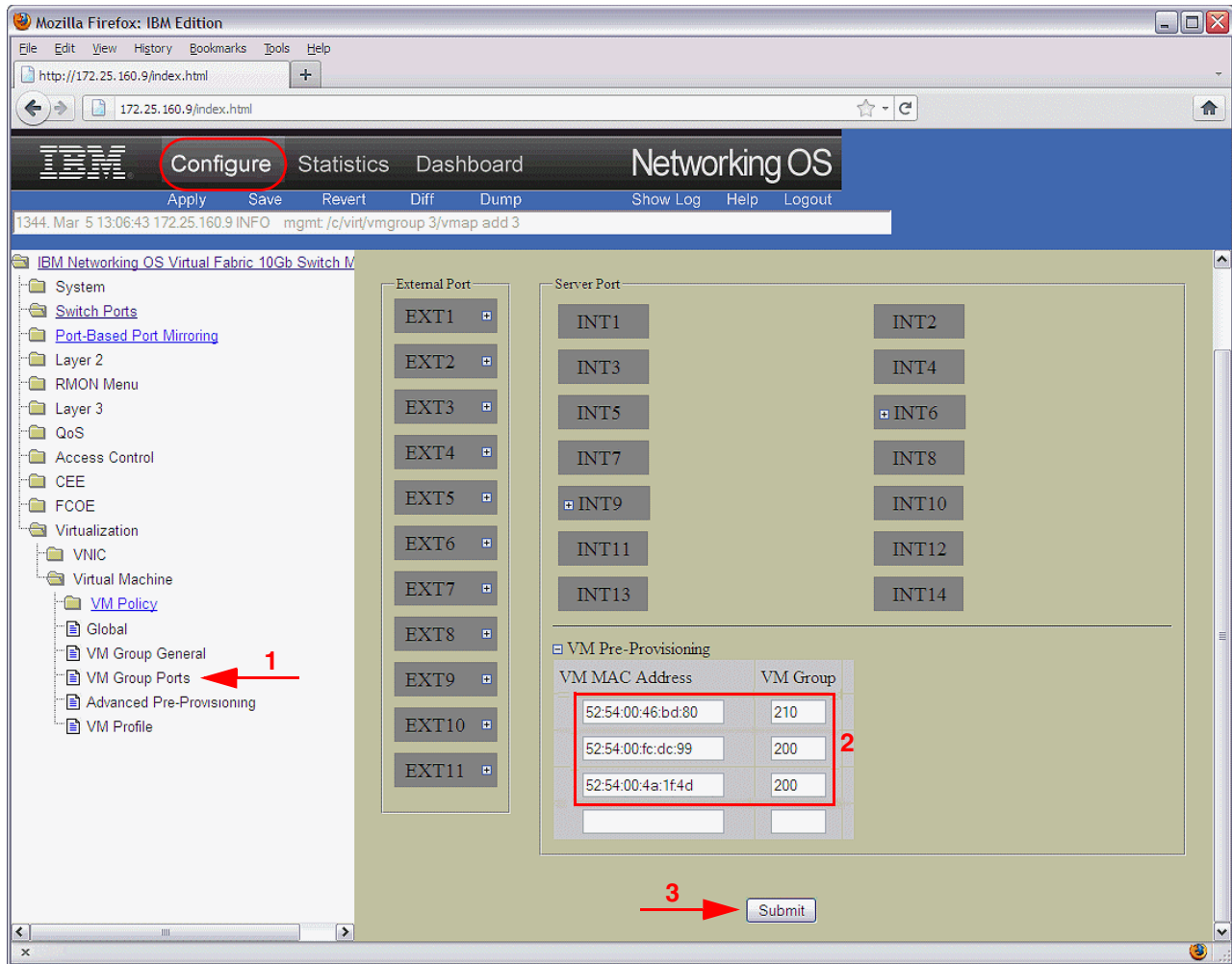


Figure 6-9 Assigning virtual machines to a VM group

To assign a virtual machine to the VM Group by using the CLI, use one of the following commands. Replace the MAC address shown in the example with one of your virtual machines.

- ▶ `/cfg/virt/vmgroup 200/addvm 52:54:00:fc:dc:99 (ibmnos)`
- ▶ `virt vmgroup 200 vm 52:54:00:fc:dc:99 (ISCLI)`

After assigning virtual machines VM groups, you can view them again by clicking the **Configure** tab and clicking **Virtualization** → **Virtual Machine** → **VM Group Ports** (Figure 6-10).

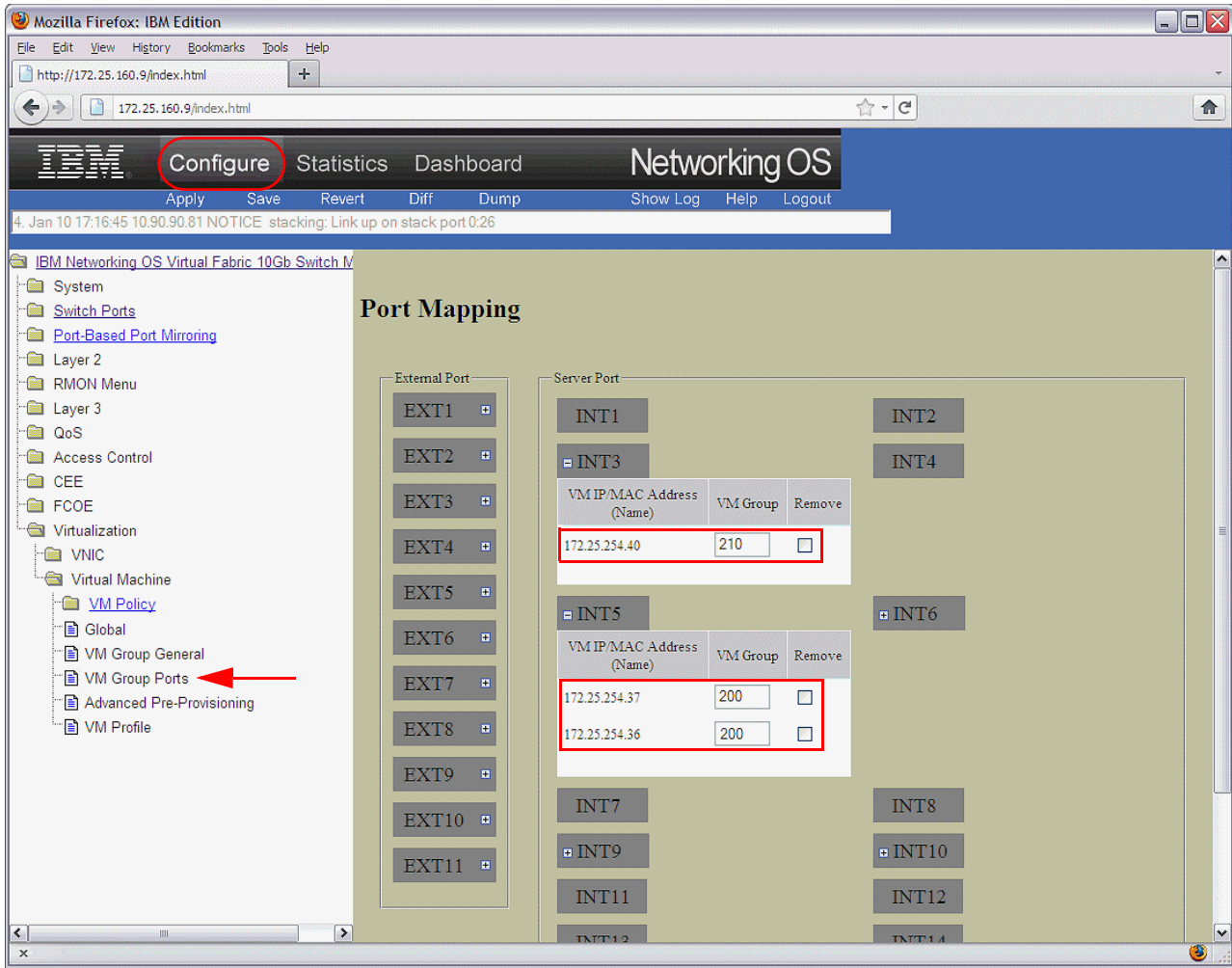


Figure 6-10 Viewing virtual machines in their VM groups

Add the physical port to which the client PC is attached to the VM group 210. This client PC is assigned to VLAN210 as shown in Figure 6-11.

Explanation: Think of VM Groups as enhanced VLANs that are virtual machine aware.

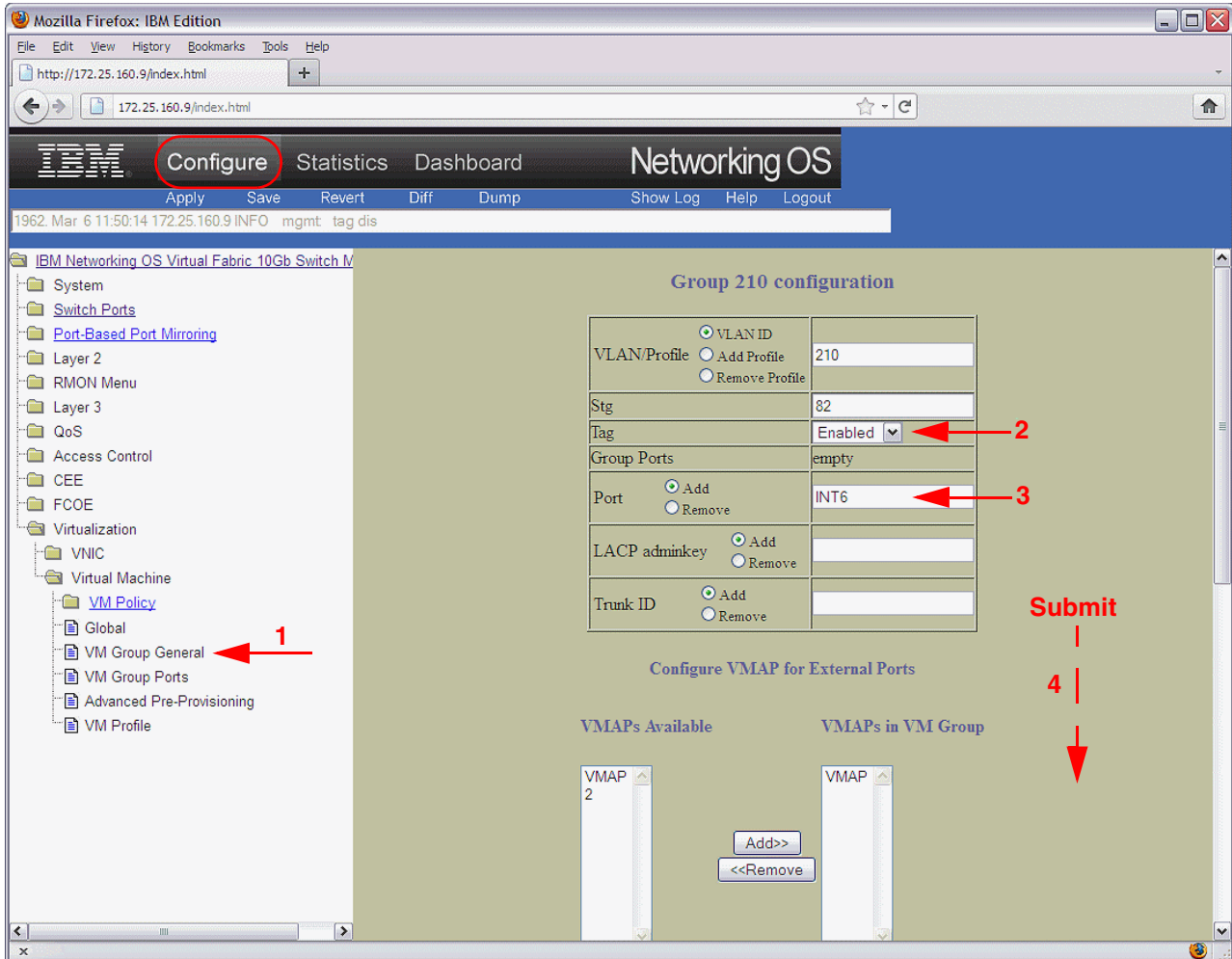


Figure 6-11 Assigning the Client PC switch port to VM Group 210

With the CLI, enter the command `/cfg/virt/vmgroup 210/port/add INT6 (ibmnos)` or `virt vmgroup 210 port INT6 (ISCLI)`.

6.2.4 Step 4: Creating a VMAP to enforce an Access Control List

Create a VMAP to enforce an ACL that prevents any communication to TCP port 80 in VM Group 200.

The WEB virtual machine is running a web server on both HTTP and HTTPS default TCP ports (80 and 443). This scenario imagines that an organization would want to prevent all non-encrypted traffic over the default web server port (TCP 80).

To create the VMAP, in the **Configure** tab, click **Access Control** → **Vlan MAP** → **Add VMAP** as shown in Figure 6-12.

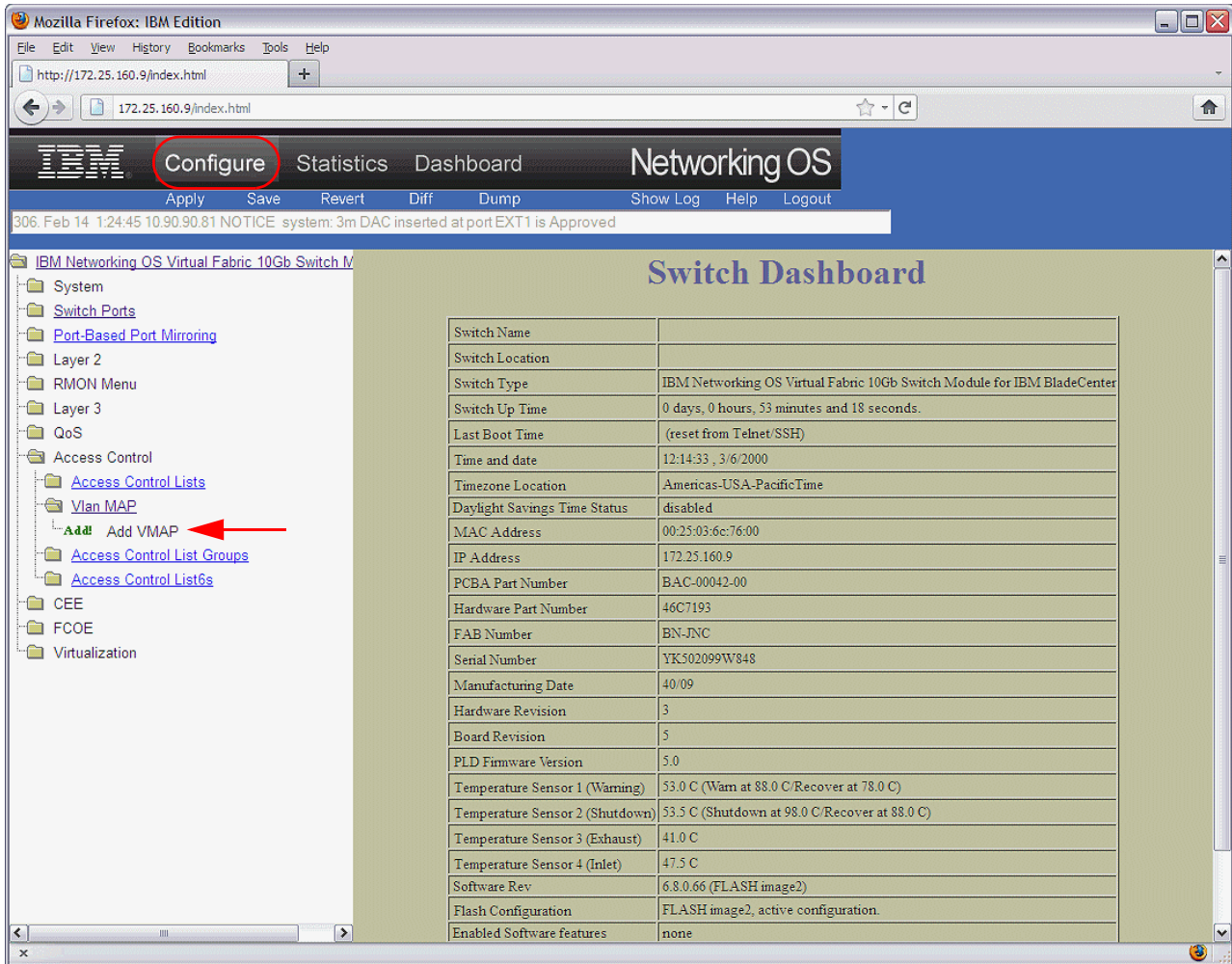


Figure 6-12 Adding a VMAP

This process opens a new window where you can configure the VMAP settings as shown in Figure 6-13. Start by giving an id to your VMAP, then select a filter action (in this example, it is **deny**). Then enter the destination port number you want to block (in this case **80**) and enable it. Then scroll down and click **Submit**, **Apply**, and then **Save**.

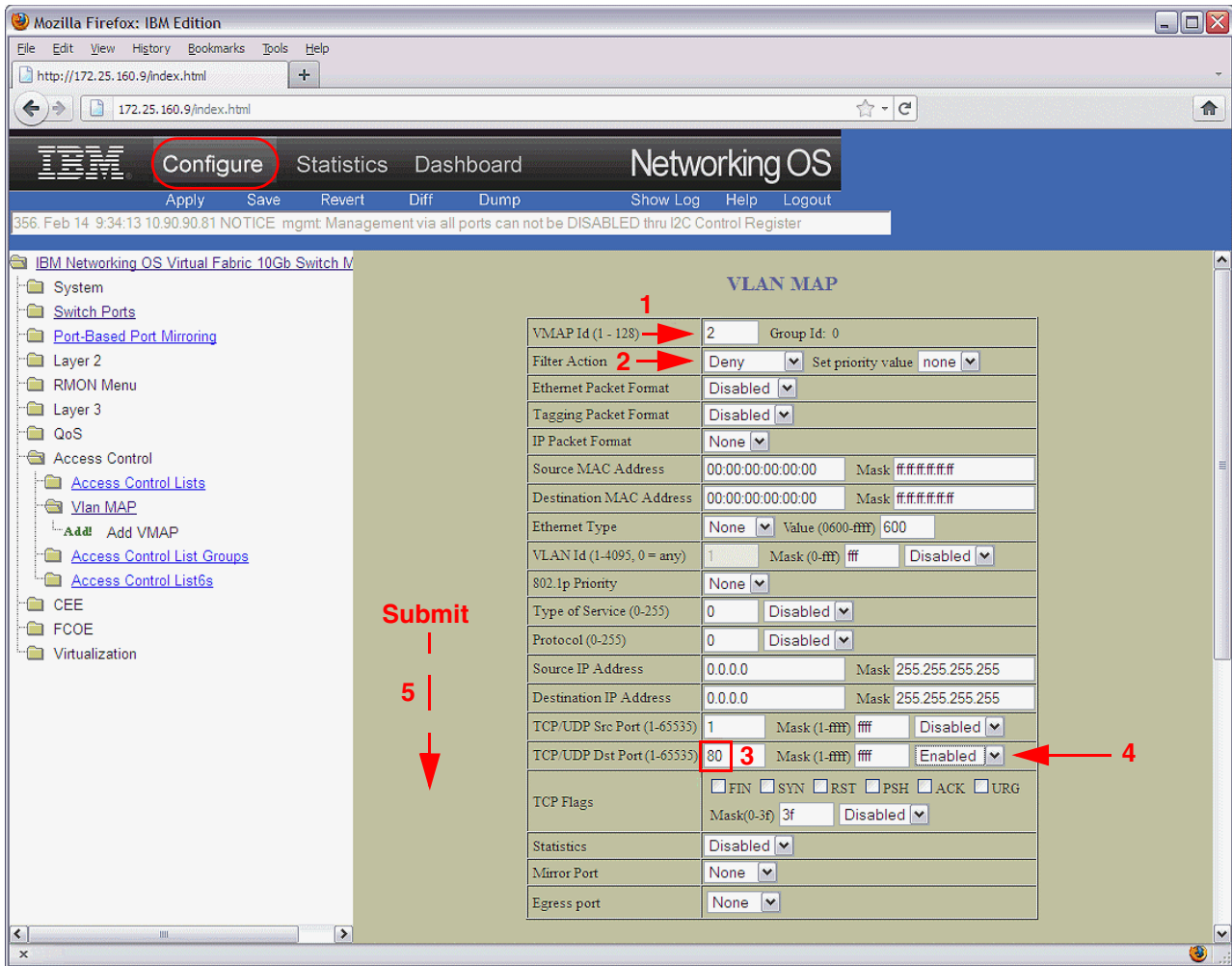


Figure 6-13 Configuring the VMAP

To create a VMAP by using the CLI, use the following commands:

- ▶ `/cfg/ac1/vmap 20/action deny` and `/cfg/ac1/vmap 20/tcpudp/dport 80 0xffff (ibmnos)`
- ▶ `access-control vmap 22 action deny` and `access-control vmap 22 tcp-udp destination-port 80 0xFFFF (ISCLI)`

Now that you created the VMAP, you need to assign it to a VM Group.

In the **Configure** tab, select **VM Group General** then select your VM Group as shown in Figure 6-14.

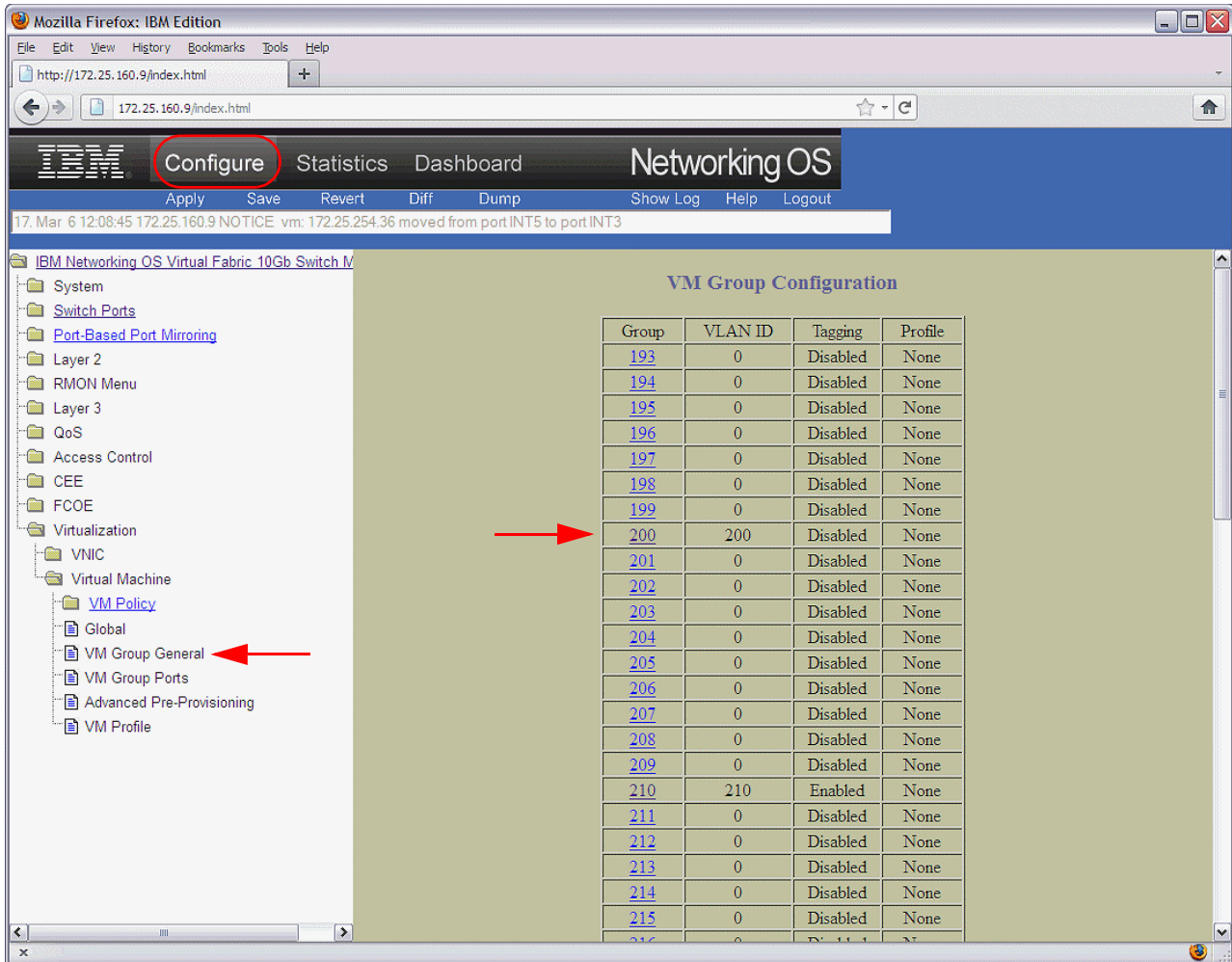


Figure 6-14 Editing VM Group 200

Then as shown in Figure 6-15, scroll down, and under **Configure VMAP for All Ports** select **VMAP 2** and click **Add**. Then click **Submit**, **Apply**, and then **Save**.

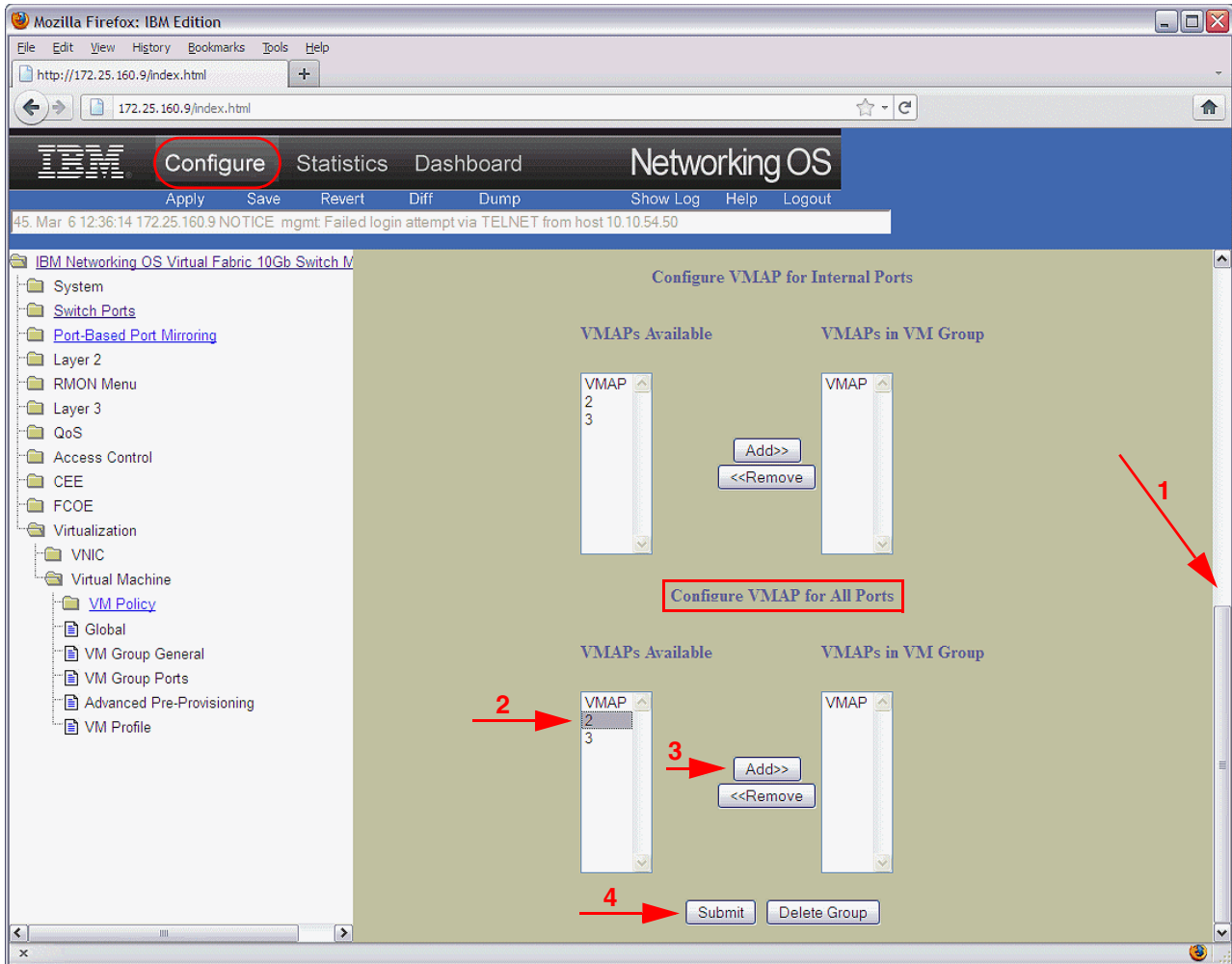


Figure 6-15 Assigning VMAP 2 to VM Group 200

To perform this step from the CLI, enter the command `/cfg/virt/vmgroup 200/vmap add 2` (ibmnos) or `virt vmgroup 200 vmap 2` (ISCLI).

On the client PC, you can see that you are not able to access the WEB virtual machine over HTTP port 80 as shown in Figure 6-16. This client can be either put in VLAN 200 or have traffic routed between VLAN 200 and VLAN 210.

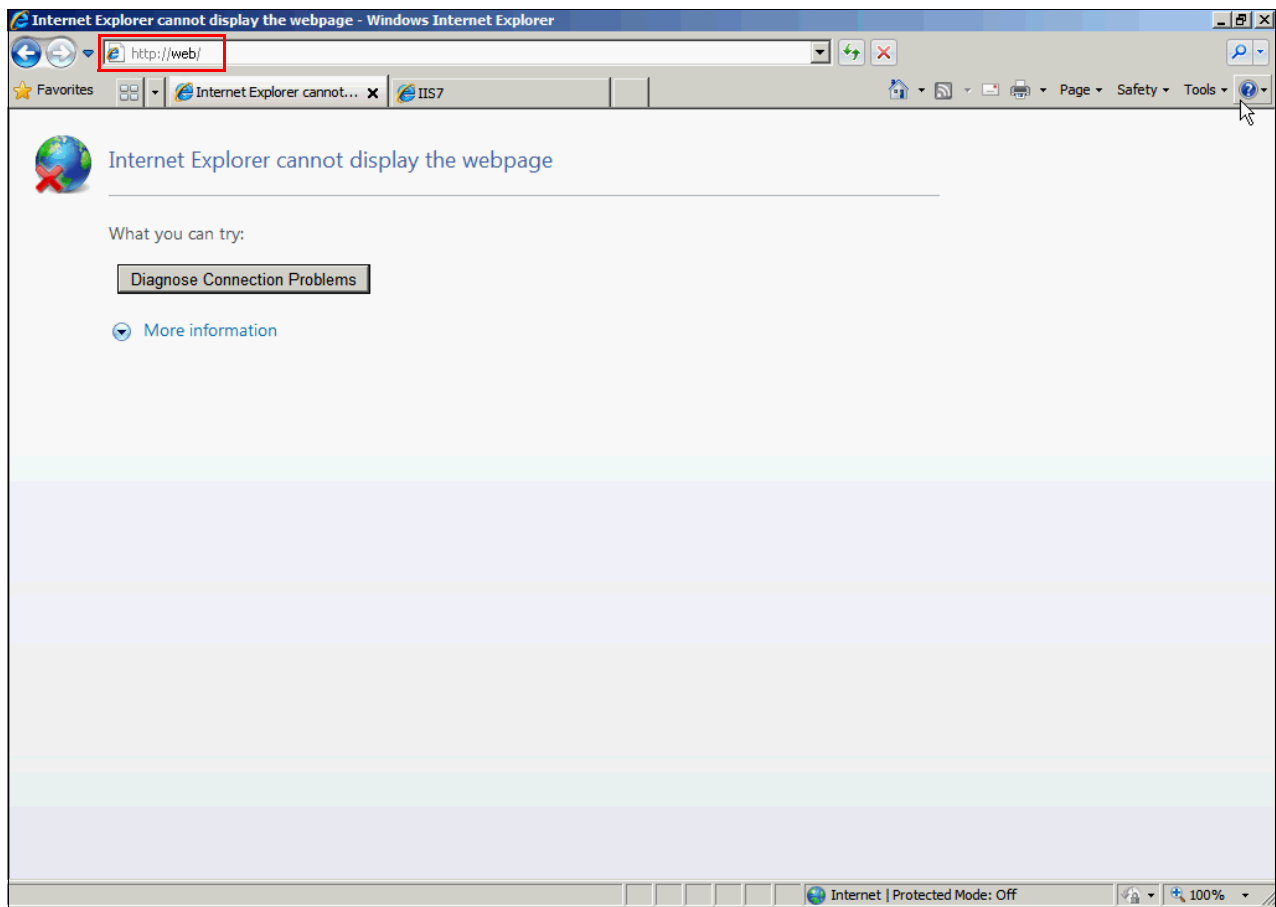


Figure 6-16 VMAP 2 is now blocking HTTP

You can still able to access that WEB virtual machine over HTTPS because only TCP port 80 is blocked in VMAP 2 (Figure 6-17).

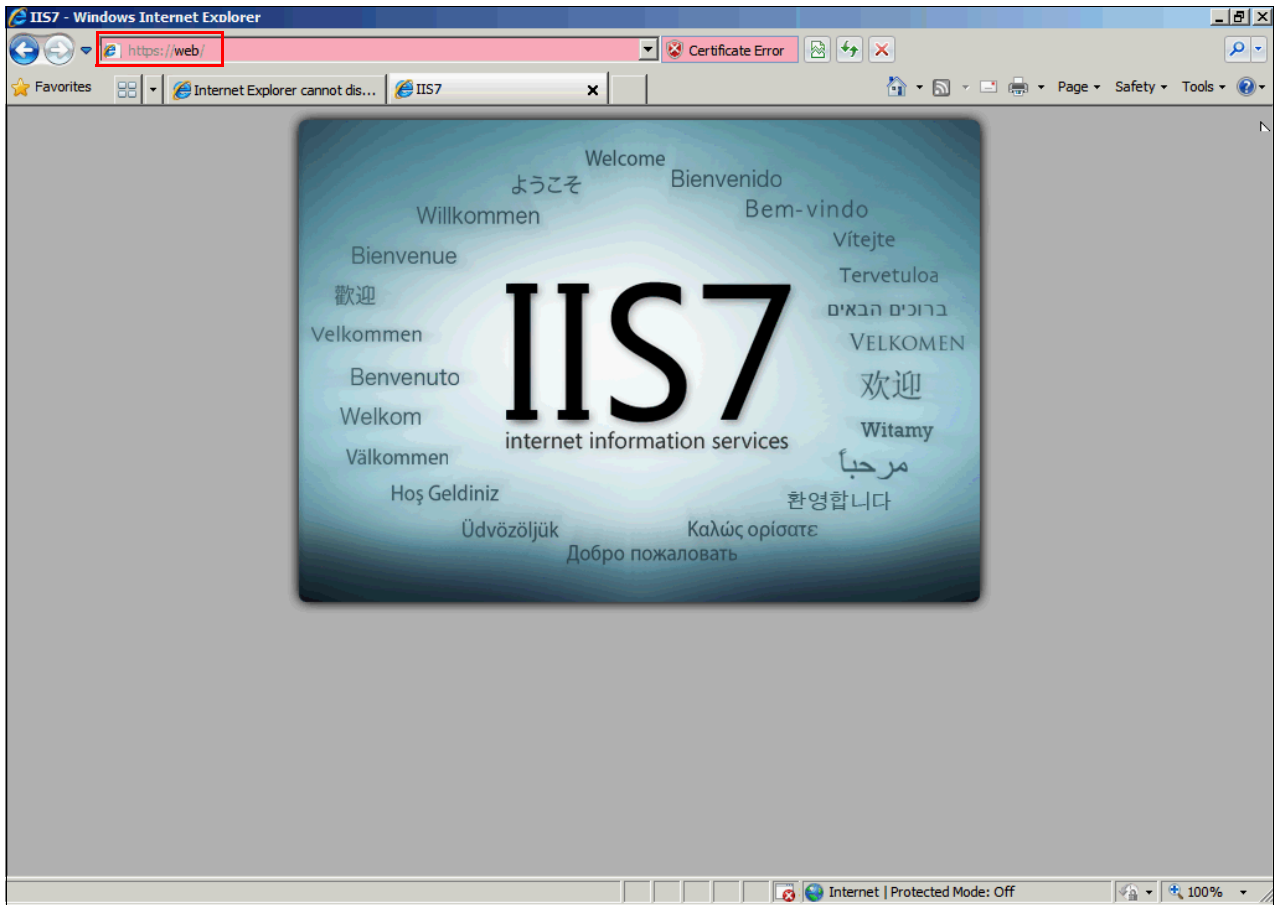


Figure 6-17 The client PC can still access HTTPS

6.2.5 Step 5: Adding a virtual machine bandwidth policy

Next, create a virtual machine bandwidth policy that limits the transmit and receive rates of the FILE virtual machine. In the **Configure** tab, click **Virtualization** → **Virtual Machine** → **VM Policy** and click **Add VM Policy**.

As shown in Figure 6-18, enter the MAC address of the virtual machine to which you want to assign this bandwidth policy. Select **Enabled** in **Bandwidth Control**, then enter a receive and transmit rate, a receive and transmit maximum burst size, and an ACL id. Then click **Submit** and **Apply**.

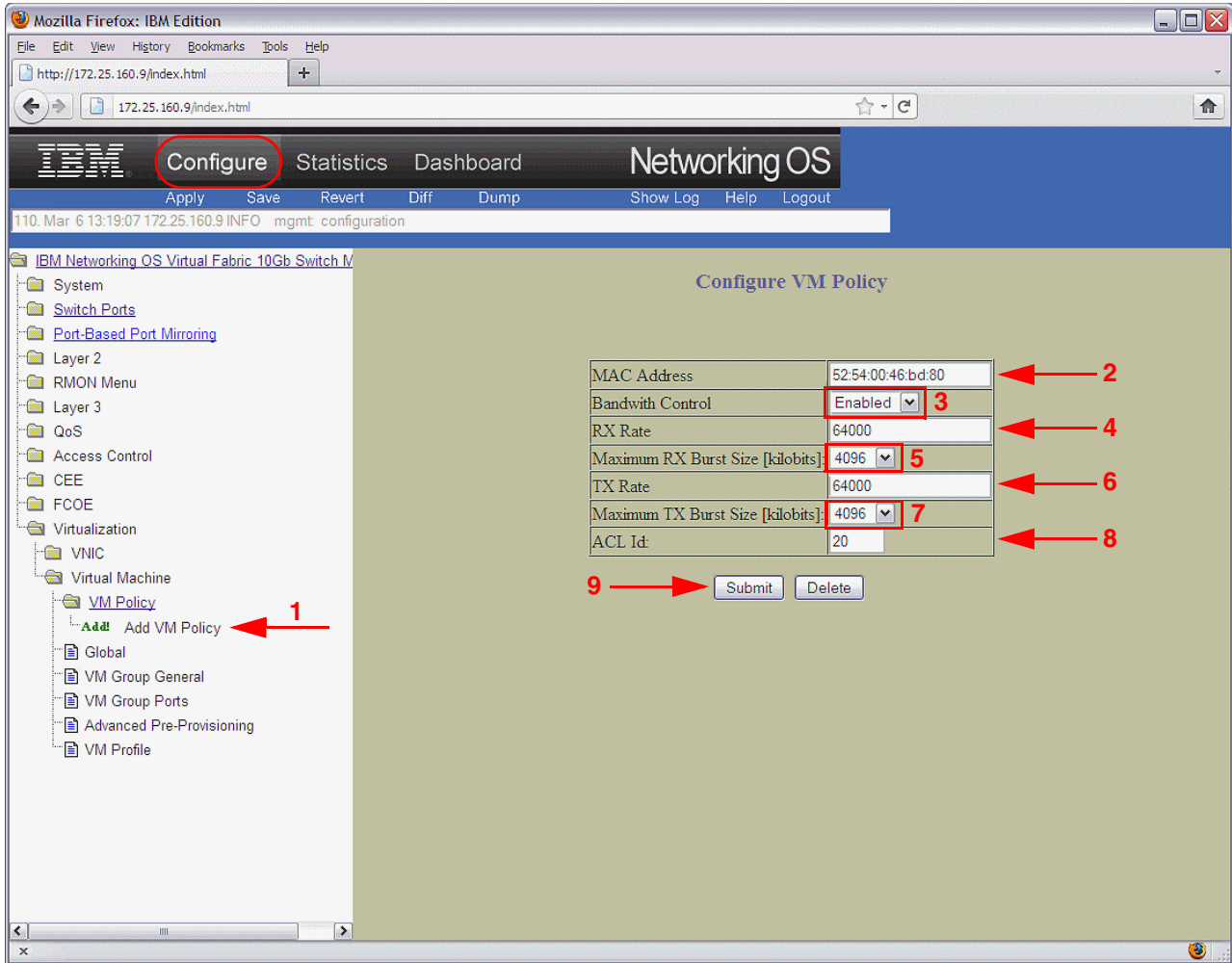


Figure 6-18 Creating a virtual machine bandwidth policy

Example 6-4 shows how to do this process from the ibmnos CLI.

Example 6-4 Creating a virtual machine bandwidth policy by using the ibmnos CLI

```
>> Main# /cfg/virt/vmpolicy/vmbwidth 52:54:00:46:bd:80
-----
[VM 52:54:00:46:bd:80 Bandwidth Management Menu]
  txrate  - Set VM Transmit Bandwidth (Ingress for switch)
  rxrate  - Set VM Receive Bandwidth (Egress for switch)
  bwctrl  - Enable/Disable VM Bandwidth Control
  delete  - Delete VM bandwidth control Entry
  cur     - Display current VM bandwidth configuration

>> VM 52:54:00:46:bd:80 Bandwidth Management# txrate 64000
Enter Max Burst Size: [32|64|128|256|512|1024|2048|4096] kilobits: 4096
Enter acl id for this meter [or hit return to choose automatically]: 20
```

```
>> VM 52:54:00:46:bd:80 Bandwidth Management# rxrate 64000
Enter Max Burst Size: [32|64|128|256|512|1024|2048|4096] kilobits: 4096
```

```
>> VM 52:54:00:46:bd:80 Bandwidth Management# bwctrl e
Current VM BW Control: disabled
New VM BW Control:     enabled
```

apply

The bandwidth policy is enabled by using JPerf. You can see in Figure 6-19 that the available network bandwidth from the client PC to the FILE virtual machine has been reduced.

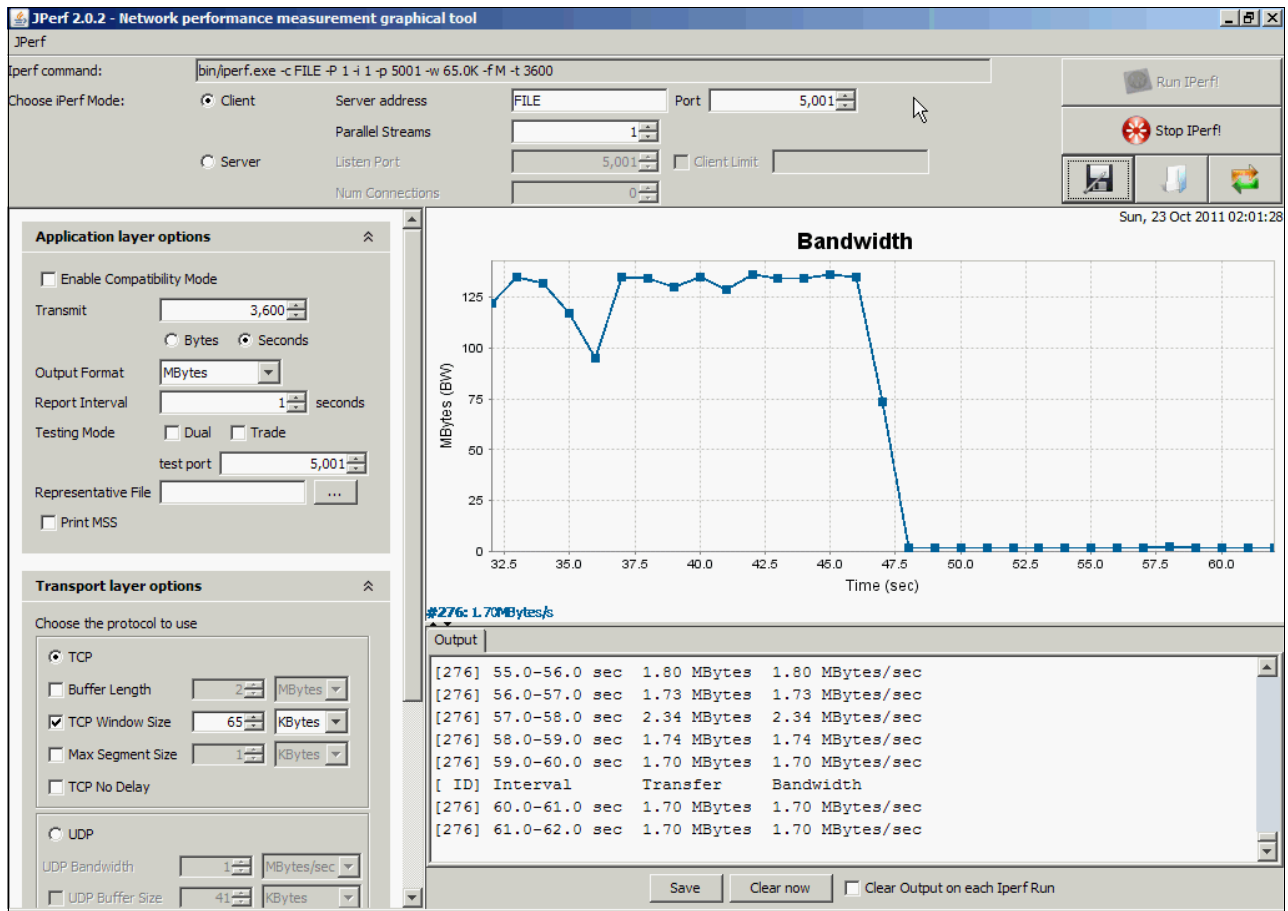


Figure 6-19 The effect of a virtual machine bandwidth policy by using JPerf

6.2.6 Step 6: Demonstrating NMotion

To demonstrate the NMotion capabilities of VMready, migrate the WEB virtual machine from the kvm2 host to the kvm1 host. Use Live Migration as shown in Figure 6-20.

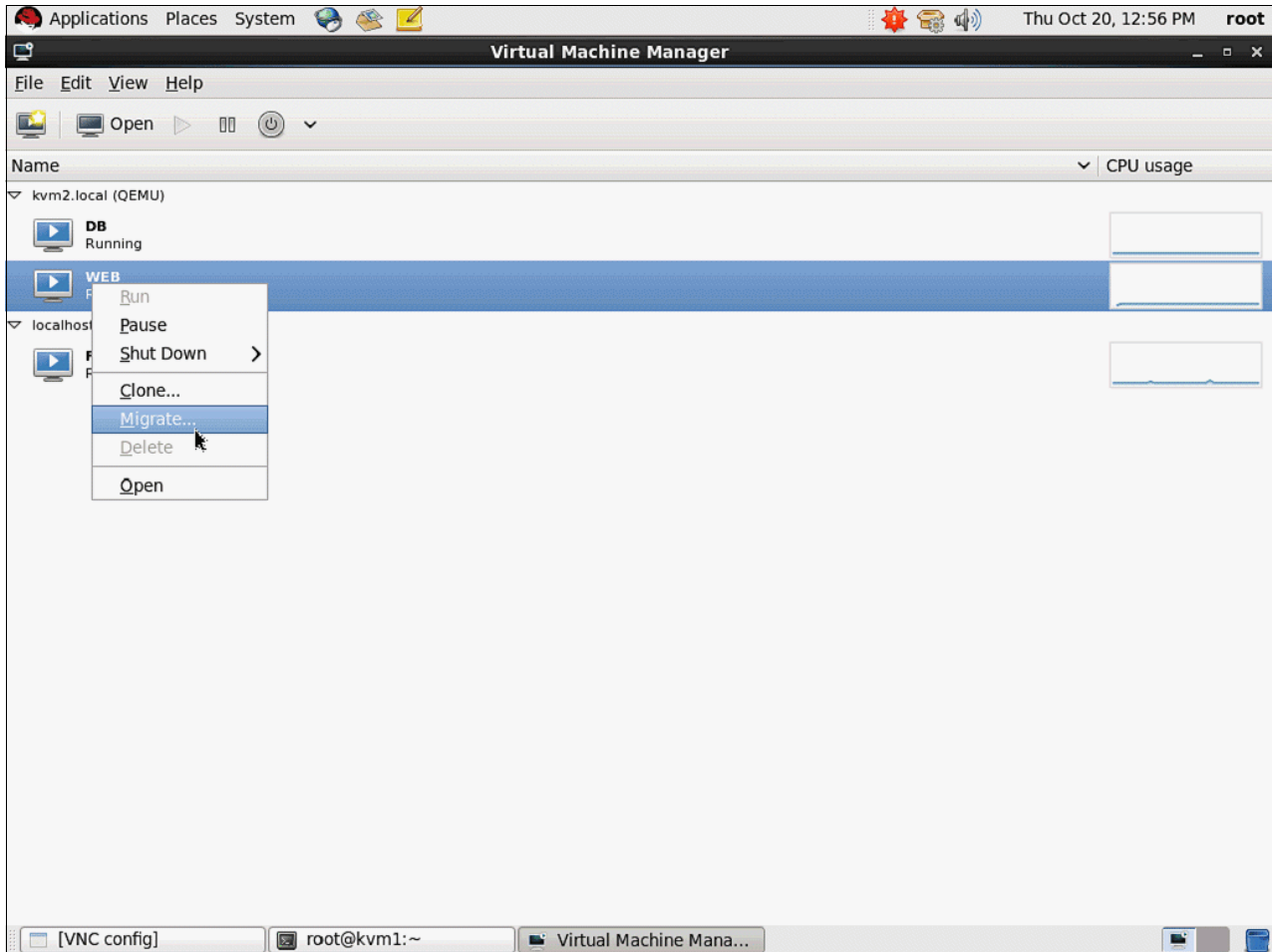


Figure 6-20 Migrating the WEB virtual machine from kvm2 to kvm1 by using Virtual Machine Manager

The results of the migration are shown in Figure 6-21.

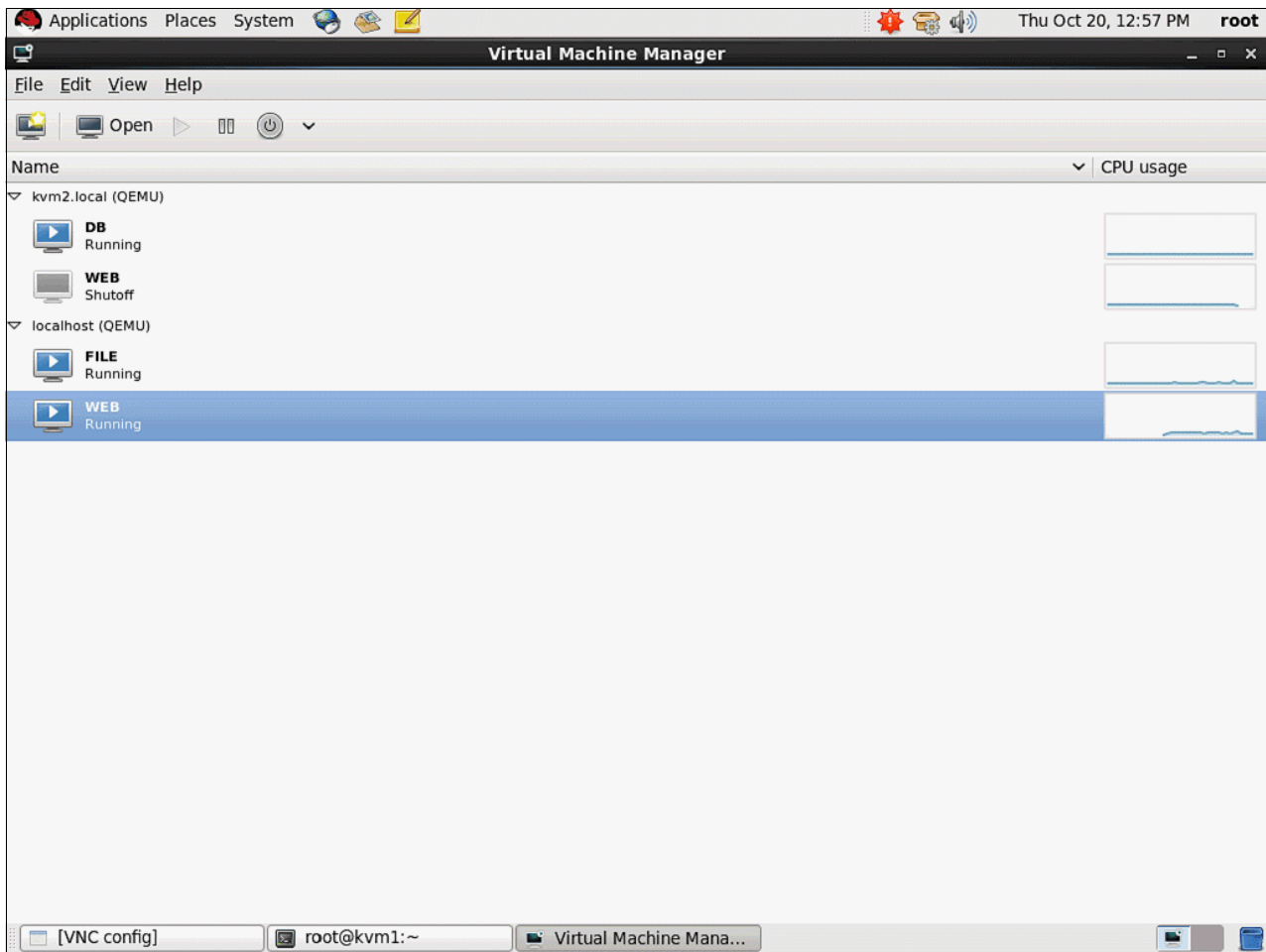


Figure 6-21 The WEB virtual machine has been migrated to kvm1

The WEB virtual machine is now migrated to the kvm1 host. You can see in the physical switch that this change has been captured as shown in Figure 6-22.

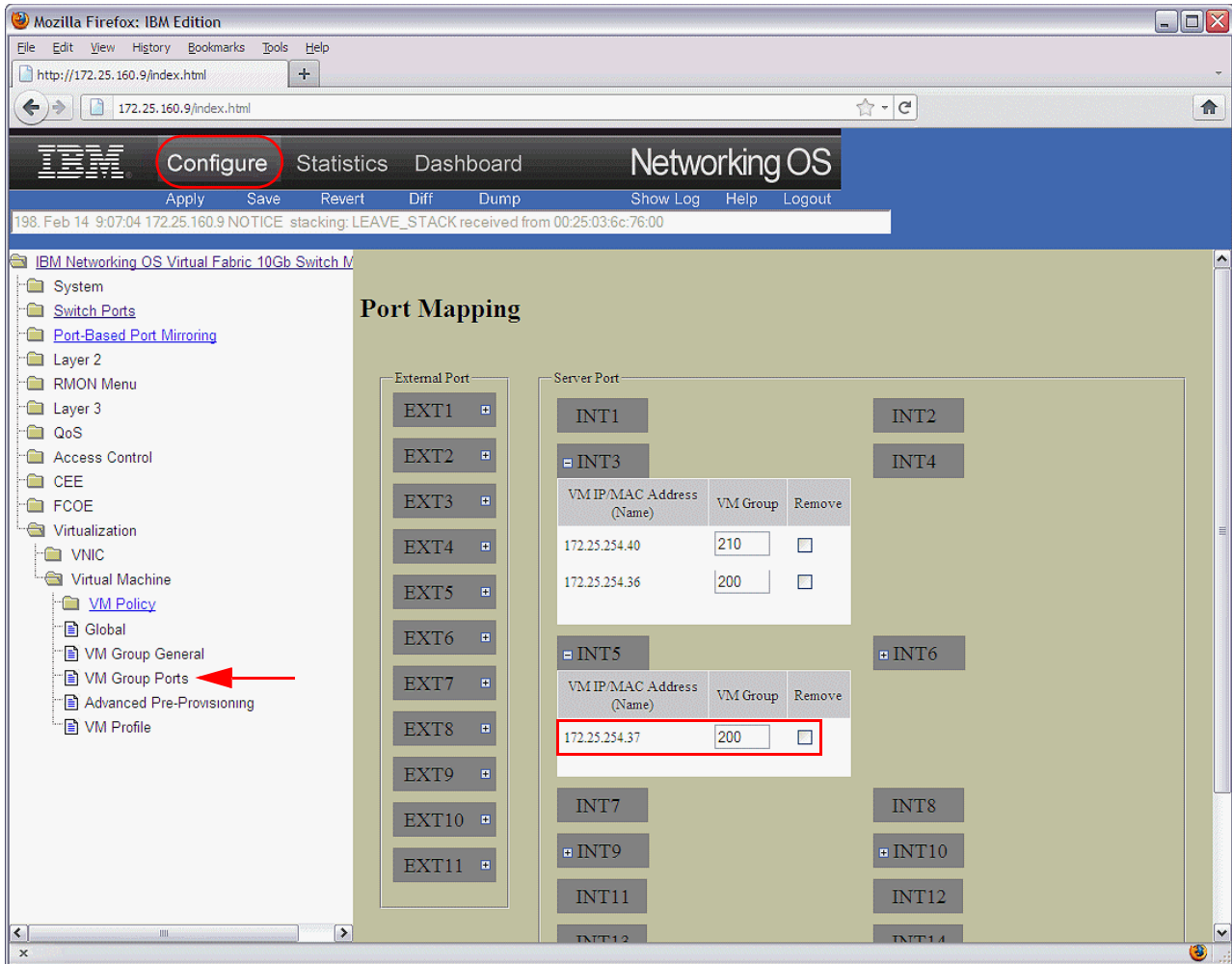


Figure 6-22 The WEB virtual machine has migrated to kvm1 in the switch GUI

You can also see that the INT3 port (to which kvm1 is attached) has been dynamically configured with VLAN 200 as shown in Example 6-5.

Example 6-5 Showing that INT3 has been dynamically configured with VLAN 200

```
>> Main# /info/port 3,5
Alias  Port Tag  Type  RMON Lrn Fld PVID  NAME  VLAN(s)
-----
INT3   3    y  Internal  d  e  e   1  INT3   1 200 210 4095
INT5   5    y  Internal  d  e  e   1  INT5   1 200 4095
```

* = PVID is tagged.

In addition, you can see that the Client PC still cannot access the WEB virtual machine over HTTP as shown in Figure 6-23.

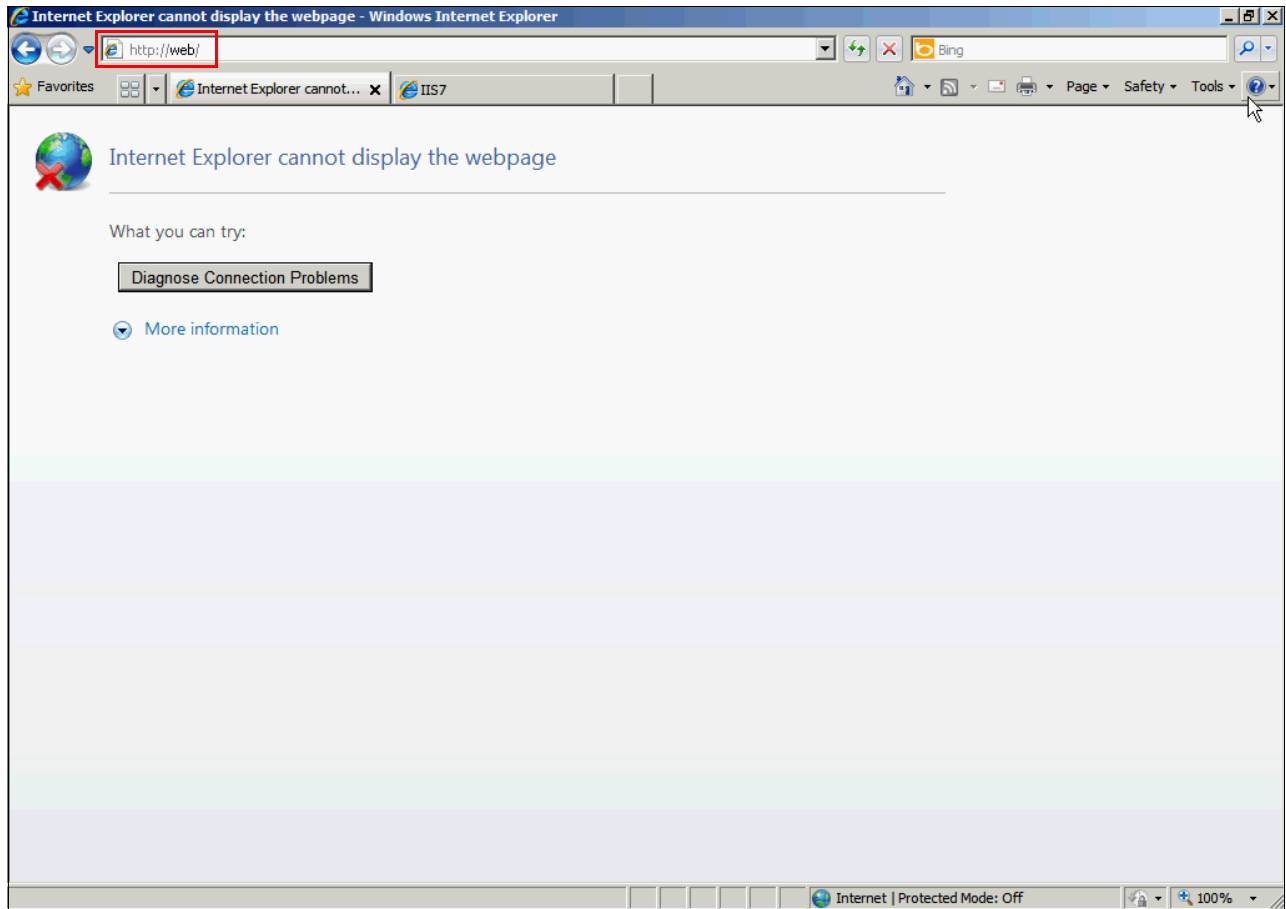


Figure 6-23 Client PC cannot access the WEB virtual machine over HTTP after it has migrated to kvm1

Move the FILE virtual machine from kvm1 to kvm2 as shown in Figure 6-24.

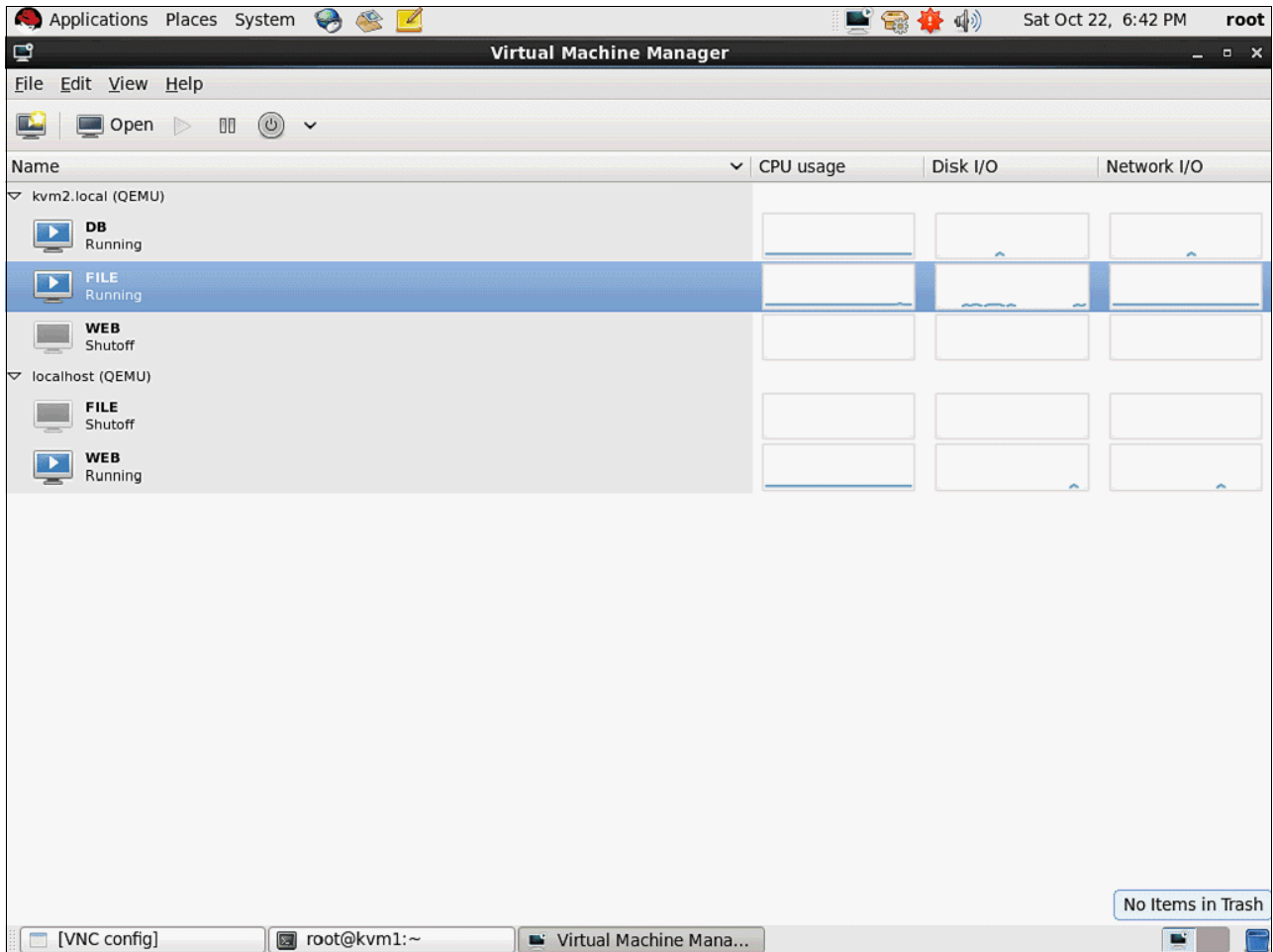


Figure 6-24 The FILE virtual machine has migrated to kvm2

Example 6-6 shows that VLAN 210 is now assigned to port INT5.

Example 6-6 VLAN 210 has moved to port INT5

```
>> Information# /info/port 3,5
```

Alias	Port	Tag	Type	RMON	Lrn	Fld	PVID	NAME	VLAN(s)
INT3	3	y	Internal	d	e	e	1	INT3	1 2 200 4095
INT5	5	y	Internal	d	e	e	2	INT5	1 2 200 210 4095

* = PVID is tagged.

Figure 6-25 shows that the virtual machine bandwidth policy is still limiting the traffic to the FILE virtual machine.

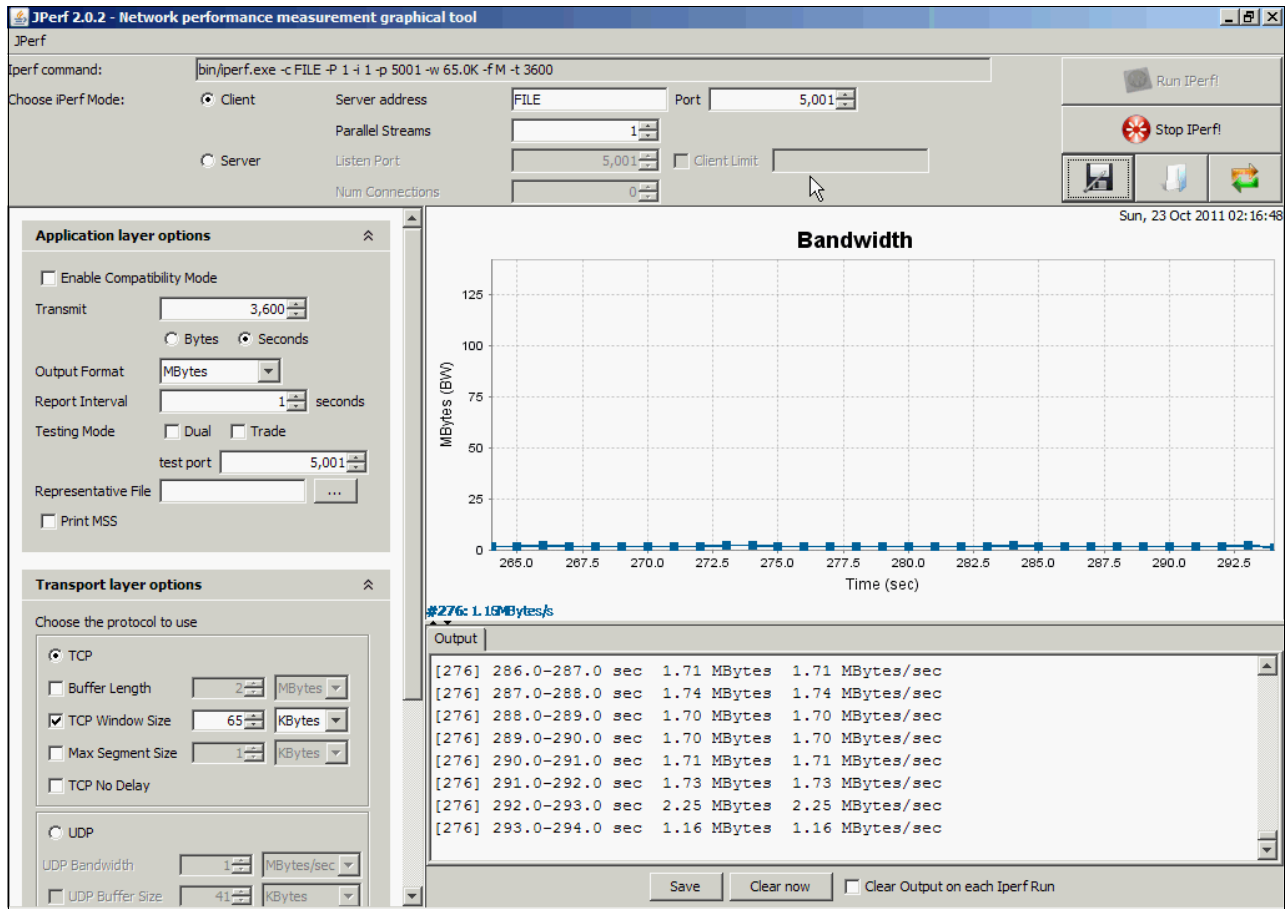


Figure 6-25 The effect of the virtual machine bandwidth policy after live migration

6.3 Implementing IEEE 802.1Qbg with VMready

802.1Qbg requires support in both the hypervisor and the physical switches.

Remember: For the rest of this chapter, the 802.1Qbg standard is called EVB (Edge Virtual Bridging).

To implement EVB support in VMready with KVM, you must update your setup with the following software:

- ▶ IBM Network Operating System v6.9.1: Version 6.9 is the IBM NOS release that supports EVB. It introduces new CLI commands in the virt menu that enable EVB configuration. IBM NOS can be downloaded from the IBM Fix Central website.
- ▶ Ildpad-0.9.43-10.el6.x86_64: The Ildpad module enables the Link Layer Discovery Protocol (LLDP), which is needed to associate virtual machines with their networking profile. Ildpad sends the required packets to the physical switch on behalf of the virtual machines. The switch in turn retrieves the networking profile, including things like the VLAN ID, ACLs, QoS settings, and traffic shaping policies, from the Virtual Server Infrastructure (VSI) database.

- ▶ virt-manager-0.9.0-7.el6.x86_64: This updated version of the Virtual Machine Manager graphical tool facilitates virtual machine networking configuration for EVB by adding the necessary configuration fields.

Restriction: At the time of writing, the lldpad module version was not fully functional. If lldpad or the physical switch is restarted while virtual machines are powered on, the VSI association is lost. The association cannot be recovered without restarting the KVM domains using virsh or virt-manager, or running a manual association with lldptool.

Figure 6-26 shows the target configuration for the lab setup. The virtual machines are now attached to a direct interface and have a VSI Type id. In addition, an IBM System Networking Element Manager 6.1 server was added that you will configure as the VSI database server (or VSI manager).

Because the virtual machines need to have direct attached network interfaces, rename the physical interfaces to match between kvm1 and kvm2. The virtual machine network is now provided through eth0 on both hosts so that you can do live migration between kvm1 and kvm2.

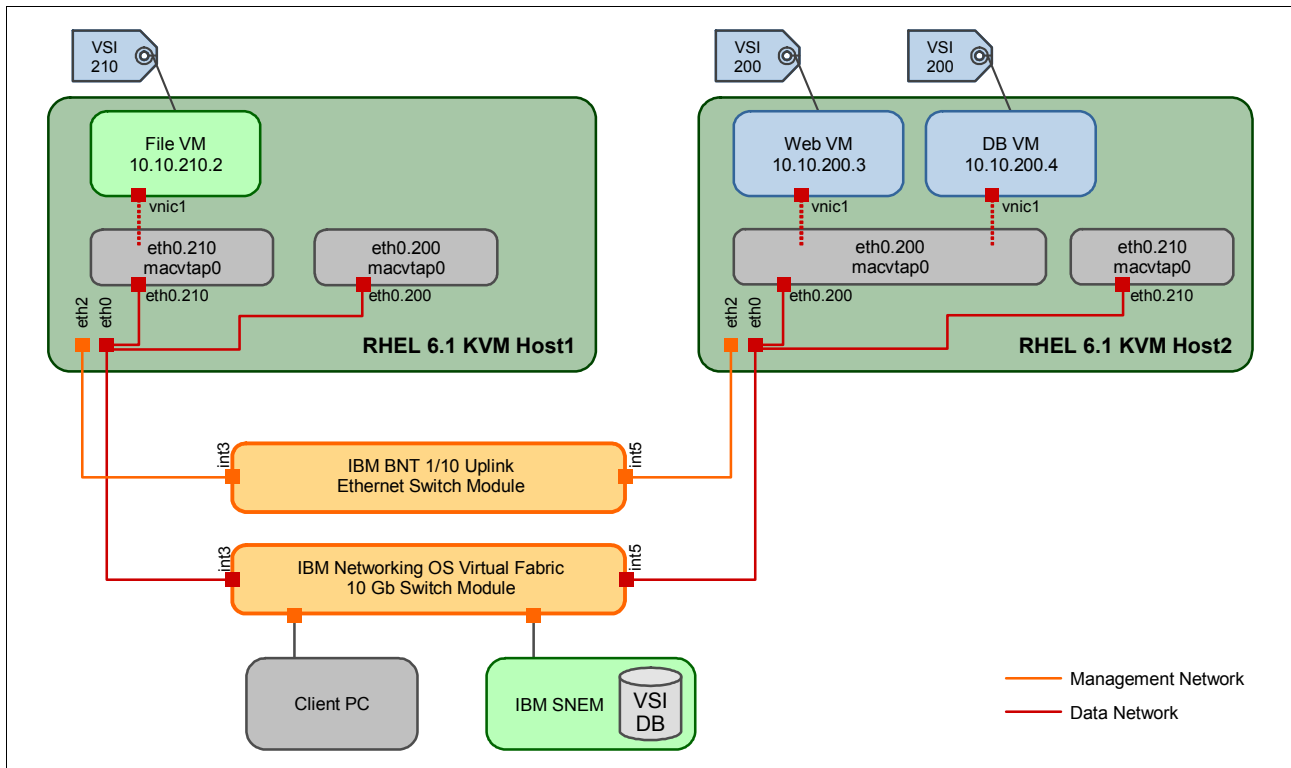


Figure 6-26 Target lab configuration for EVB

Figure 6-27 shows the necessary configuration steps on the different components of the solution.

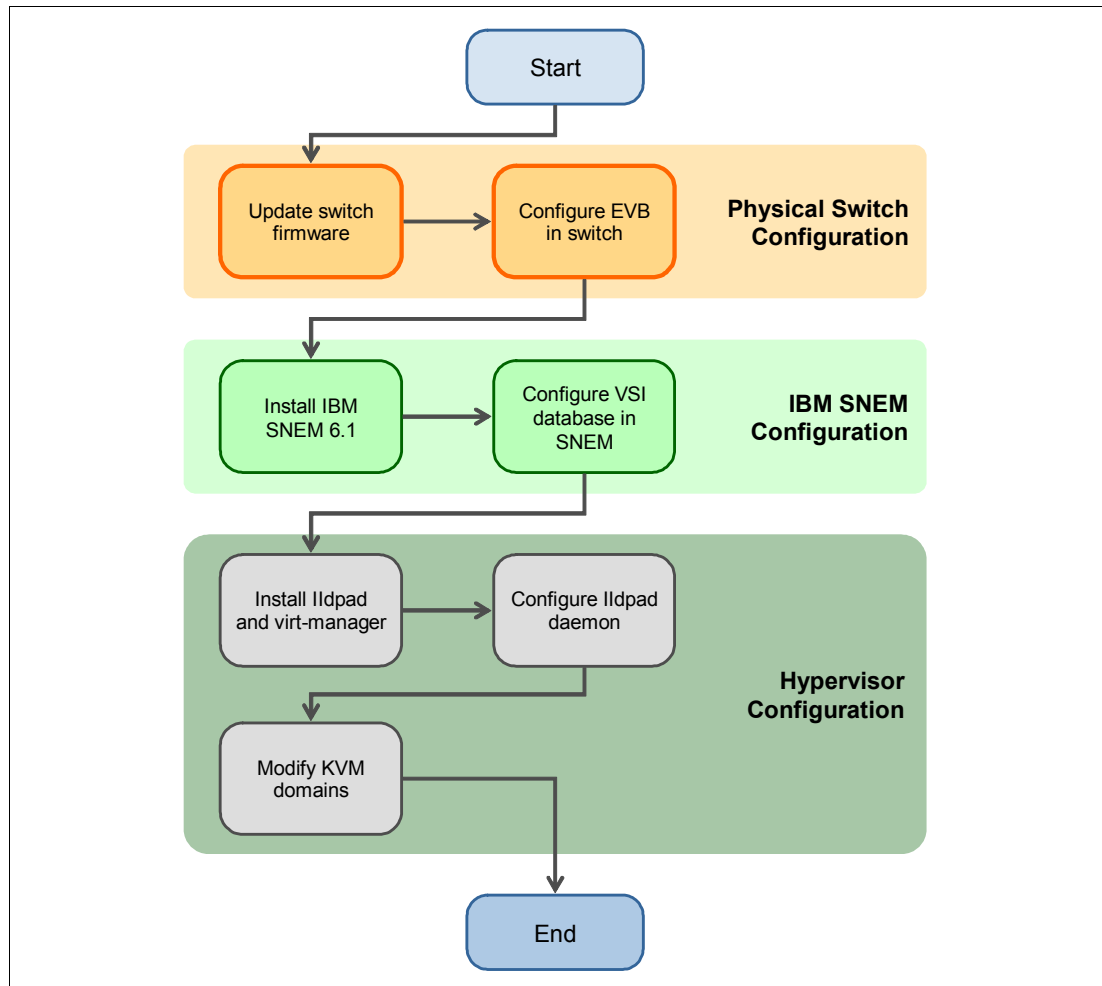


Figure 6-27 EVB configuration workflow

Tip: The order of the configuration steps only matters if you want to minimize outage while enabling EVB. In this case, you would want to use new uplinks for your virtual machine networking and start with the VSI database configuration.

6.3.1 Step 1: Configuring the physical switch

The following steps are necessary to configure EVB support in the physical switch:

1. Update the switch firmware to version 6.9.1 or higher. This can be done from the CLI, which requires a TFTP server on your network. It can also be done from the graphical web interface of the switch, in which case you can upload the firmware file by using a web browser. See the IBM NOS application guide for the exact procedure for your switch.
2. Enable LLDP on the switch with the `/cfg/12/11dp/on` command (ibmnos).

Restriction: At the time of writing, only the ibmnos can be used to configure EVB on the physical switch. Configuration using ISCLI or the BBI is not functional with IBM Networking Operating System 6.9.1.

3. Disable classical VMready on the switch with the `/cfg/virt/disvmr` command (ibmnos).
4. Create an EVB profile and enable reflective relay and VSI discovery on that profile as shown in Example 6-7.

Example 6-7 Creating an EVB profile

```
/cfg/virt/evb/profile 2

/cfg/virt/evb/profile 2/rr ena
Current Reflective Relay: disabled
New Reflective Relay:      enabled

/cfg/virt/evb/profile 2/vsidisc ena
Current VSI Discovery: disabled
New VSI Discovery:        enabled
```

5. Add the EVB profile to the switch ports where your KVM servers virtual machine network interfaces are connected as shown in Example 6-8.

Example 6-8 Adding the EVB profile to switch ports

```
/cfg/port INT5/evbprof 2
Current evb profile ID:      0
New pending evb profile ID: 2
Enabled nearest customer bridge support in LLDP.
```

6. Configure a VSI database server definition on the switch as shown in Example 6-9. The setup specifies the IP address of the System Networking Element Manager server that you will use to host your VSI database.

Example 6-9 Configuring a VSI database server definition on the switch

```
/cfg/virt/evb/vsidb 1/
-----
[VSI Type DB 1 Menu]
managrip - Set VSI DB Manager IP
port     - Set VSI DB Manager Port
docpath  - Set VSI DB Document Path
alltypes - Set VSI DB Document Path
cur      - Display current VSI Type configuration
reset    - Reset VSIDB Info

>> VSI Type DB 1# managrip 172.25.75.175
Current IP address:      0.0.0.0
New pending IP address: 172.25.75.175

>> VSI Type DB 1# port 40080

>> VSI Type DB 1# docpath "bhm/rest/vsitypes"
```

Current VSI Type DB URI Path:
New VSI Type DB URI Path: bhm/rest/vsitypes

Restriction: At the time of writing, the IBM NOS 6.9 code had a limitation that prevented EVB from working with the default STP operation mode (pvrst) under some circumstances. These circumstances included adding a VLAN on the switch for non-EVB systems such as a physical server. To work around this issue, use the `/cfg/12/mrst/mode rstp` command, which modifies the STP operation mode for the whole switch.

6.3.2 Step 2: Configuring the IBM System Networking Element Manager

IBM System Networking Element Manager version 6.1 is the first release of that product (formerly known as Blade Harmony Manager) to implement EVB support.

In addition to configuring a VSI database, IBM System Networking Element Manager provides many other functions. It can also facilitate the physical switch configuration for EVB support. For more information about IBM System Networking Element Manager, see Chapter 3, “Management for VMready” on page 23.

After installing IBM System Networking Element Manager, open the web interface by using a web browser. The default URL is `http://<SNEM server IP address>:40080/bhm`. Log in with the *admin* user (the default user name and password are *admin/admin*).

From the **Device List** window, click **Options** → **VSI DB Console** as shown in Figure 6-28.

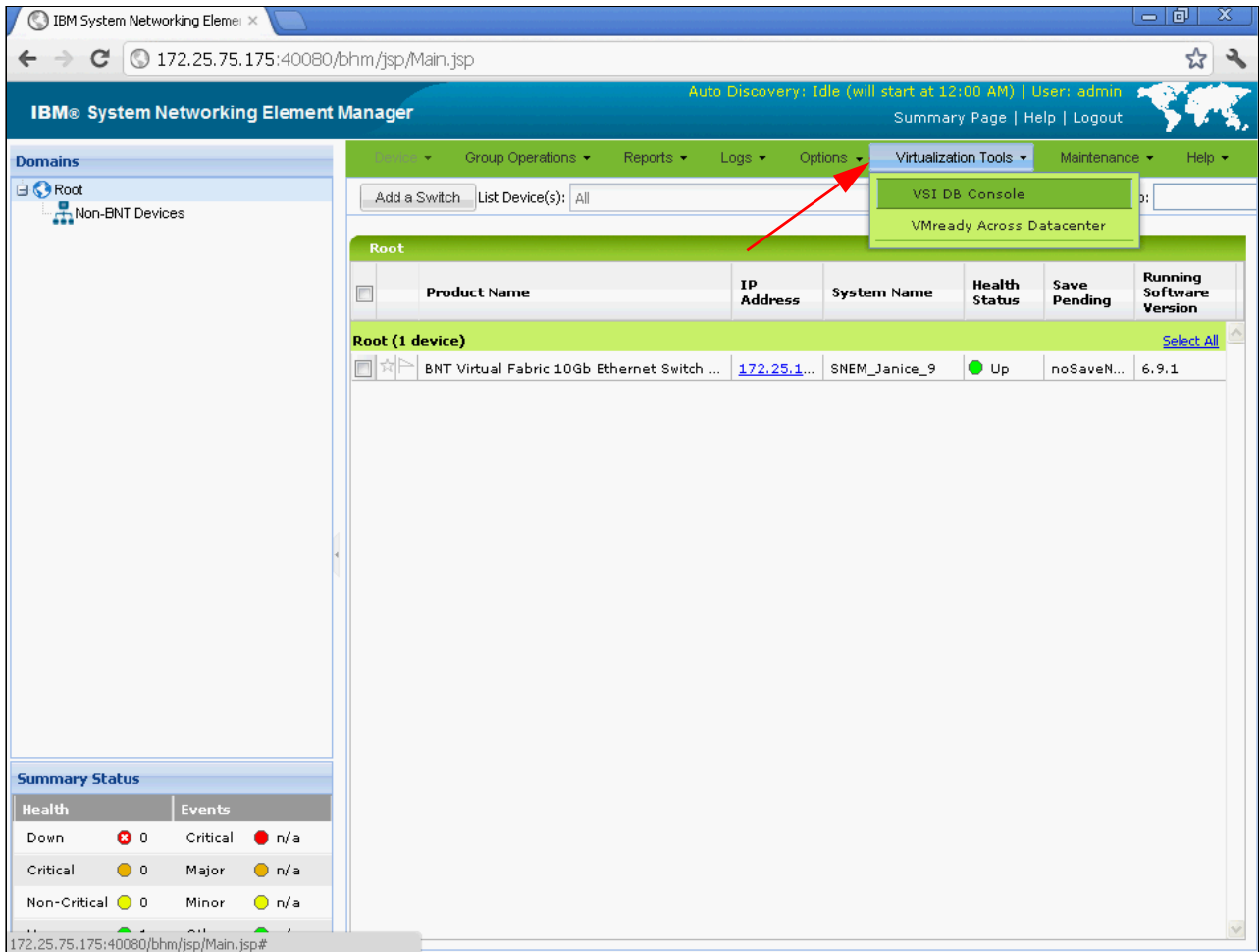


Figure 6-28 Opening the VSI DB Console in IBM System Networking Element Manager 6.1

In the VSI DB Console window, click the VSI Type tab, click **Insert**, and configure the first VSI type ID as shown in Figure 6-29.

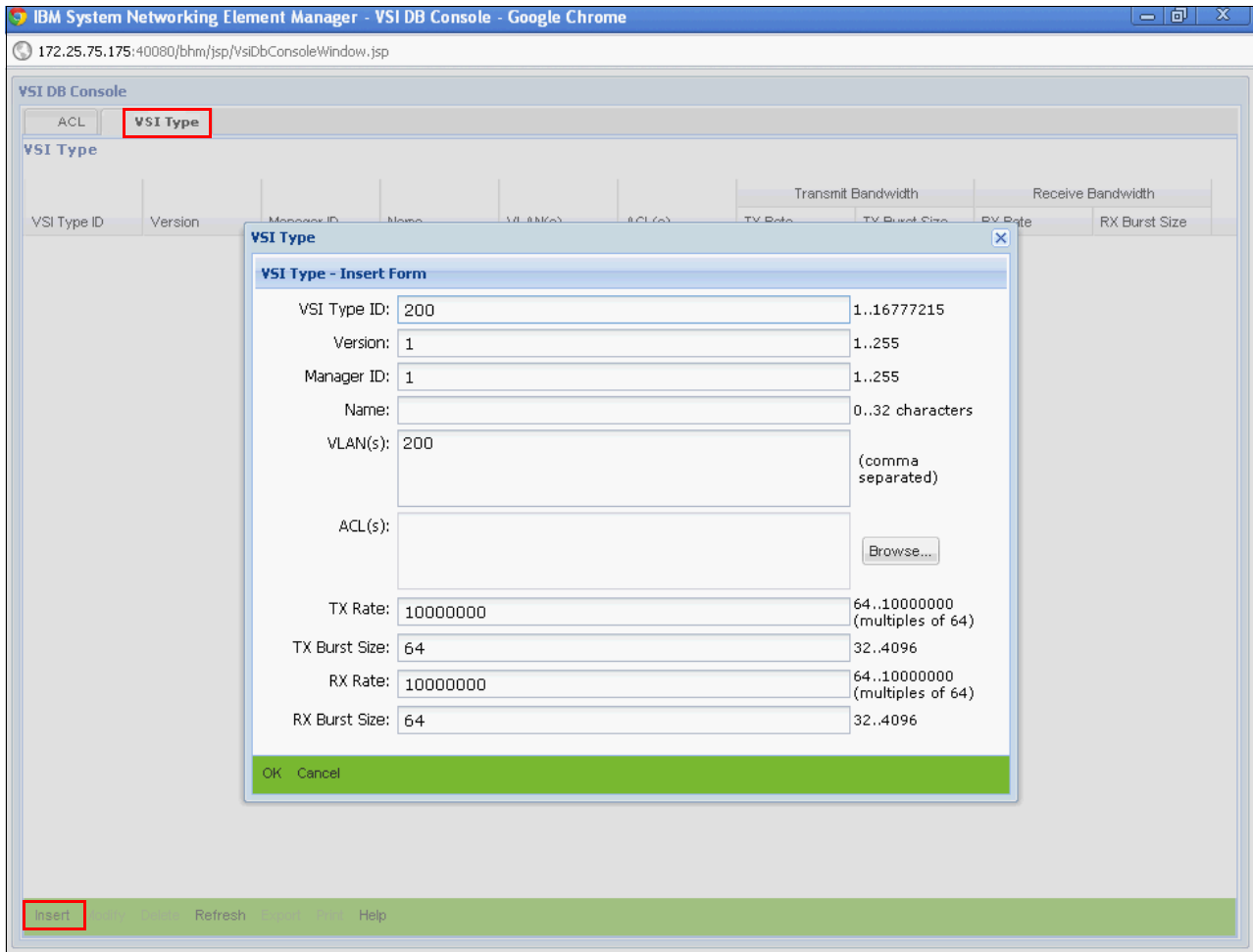


Figure 6-29 Creating a VSI type ID in IBM System Networking Element Manager 6.1

Create as many VSI type IDs as required. In this setup, create VSI type ids 200 and 210, which have VLANs 200 and 210 assigned to them.

You can verify that the VSI type IDs are accessible from the physical switch by running the commands shown in Example 6-10 on the physical switch.

Example 6-10 Verifying that the physical switch can retrieve the VSI type IDs from the VSI database

```
>> Main# /info/virt/evb/vdp/vsidb
INDEX : 1
-----
Name           : WEB
Type ID        : 200
Version        : 1
Manager ID     : 1
VLAN           : 200
TxRate         : 10000000
TxBurst        : 64
RxRate         : 10000000
RxBurst        : 64
```

INDEX : 2

Name	: FILE
Type ID	: 210
Version	: 1
Manager ID	: 1
VLAN	: 210
TxRate	: 10000000
TxBurst	: 64
RxRate	: 10000000
RxBurst	: 64

Remember: When the VSI database server is unavailable, the physical switch uses the local cache copy of the database it has in memory.

You can also verify the availability of your VSI type configuration by opening the following URL in a web browser: `http://<IBM SNEM server IP address>:40080/bhm/rest/vsitypes`. Doing so should display an XML file as shown in Figure 6-30.

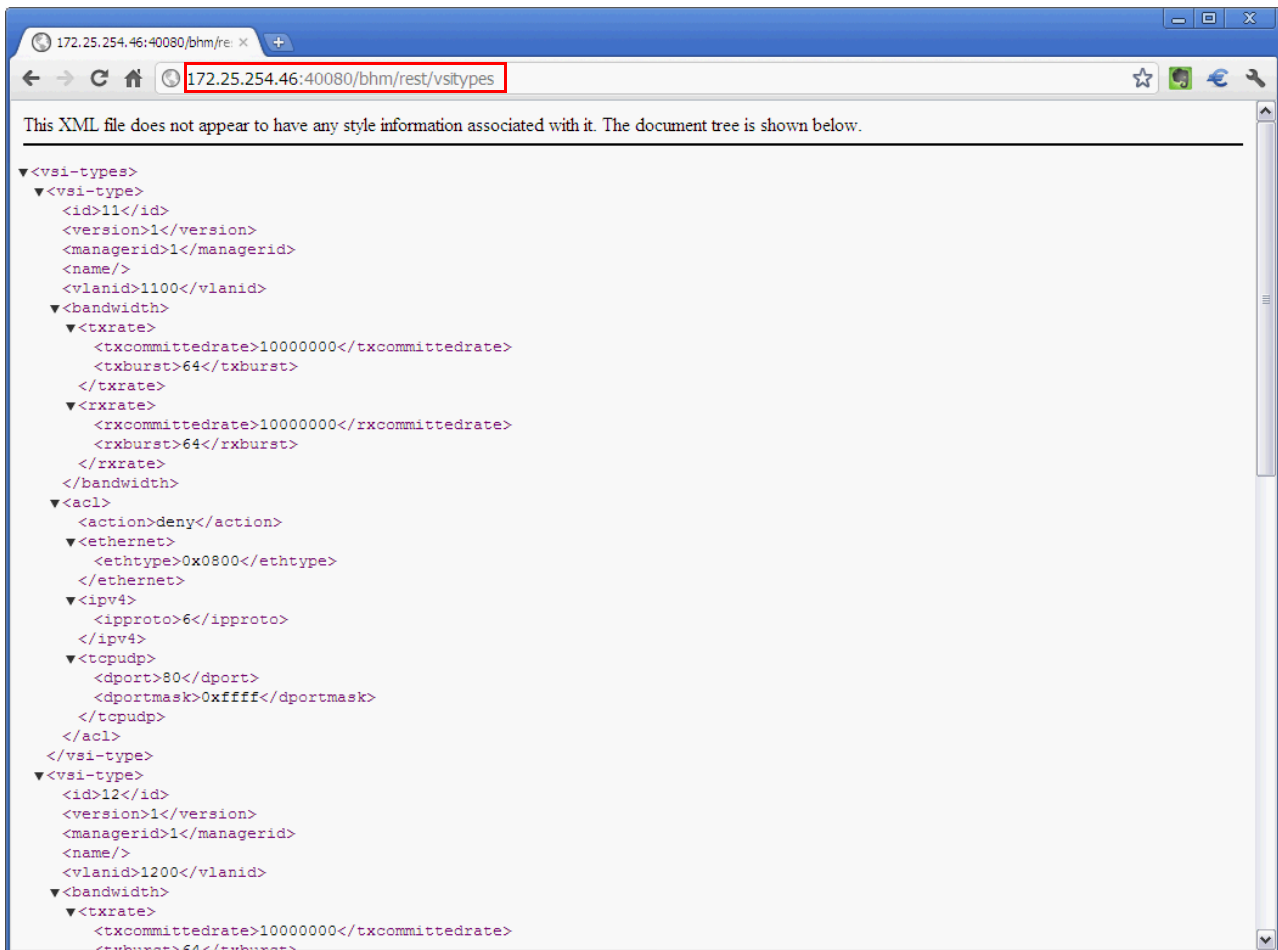


Figure 6-30 Displaying the VSI DB information in a web browser

6.3.3 Step 3: Configuring the hypervisor

The KVM hypervisor configuration requires three steps:

- ▶ Installing and configuring the lldpad package
- ▶ Updating the virt-manager package
- ▶ Configuring the KVM domains (virtual machines) network interfaces

Installing and configuring the lldpad package

To install lldpad, run the `yum install lldpad` command as root as shown in Example 6-11.

Example 6-11 Installing lldpad

```
[root@kvm1 ~]# yum install lldpad
Loaded plugins: product-id, refresh-packagekit, rhnplugin, subscription-manager
Updating Red Hat repositories.
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package lldpad.x86_64 0:0.9.41-4.el6_1.5 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
Package                Arch          Version
Repository              Size
=====
Installing:
  lldpad                x86_64
0.9.41-4.el6_1.5       rhel-x86_64-server-6
160 k
```

Transaction Summary

```
=====
Install      1 Package(s)
```

```
Total download size: 160 k
Installed size: 0
Is this ok [y/N]: y
Downloading Packages:
lldpad-0.9.41-4.el6_1.5.x86_64.rpm
| 160 kB    00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: lldpad-0.9.41-4.el6_1.5.x86_64
1/1
duration: 174(ms)
Installed products updated.
```

Installed:
lldpad.x86_64 0:0.9.41-4.e16_1.5

Complete!

If you already have lldpad installed and just need to upgrade it, use the **yum update lldpad** command. If the package you need is not available in your yum repository, use the **rpm upgrade** command as shown in Example 6-12. You need lldpad version 0.9.43-10 or later to run this command.

Example 6-12 Upgrading lldpad by using rpm

```
[root@kvm1 ~]# rpm -Uvh lldpad-0.9.43-10.e16.x86_64.rpm
Preparing...                               ##### [100%]
 1:lldpad                                   ##### [100%]
```

Configure lldpad to run when the server reboots by using the **chkconfig lldpad on** command.

Next, start the lldpad service with the **service lldpad start** command.

To configure lldpad for the network interface that is used to connect virtual machines, use the commands shown in Example 6-13. In this example, the interface is eth0.

Example 6-13 Configuring lldpad

```
lldptool -i eth0 -g ncb -T -V evbCfg -c enableTx=yes -c fmode=reflectiverelay -c
capabilities=rte,ecp,vdp
lldptool -L -i eth0 -g ncb adminstatus=rxtx
lldptool -T -i eth0 -g ncb -V vdp enabletx=yes

lldptool -i eth0 -T -V evbCfg -c enableTx=yes -c fmode=reflectiverelay -c
capabilities=rte,ecp,vdp
lldptool -L -i eth0 adminstatus=rxtx
lldptool -T -i eth0 -V vdp enabletx=yes
```

Restriction: At the time of writing, it was necessary to configure lldpad for both the nearest bridge and the nearest customer bridge as shown in Example 6-13. The 802.1Qbg standard normally specifies that only the nearest customer bridge is used. This limitation should be removed in the next release of lldpad.

The nearest customer bridge is configured by using the **-g ncb** option with **lldptool**. When **-g** is not used, **lldptool** targets the nearest bridge by default as documented in the lldptool man page.

Finally, restart the lldpad service by entering the **service lldpad restart** command.

The lldptool writes the configuration in the `/var/lib/lldpad/lldpad.conf` configuration file. In both the `lldp` and `nearest_customer_bridge` sections, you should see entries similar to those shown in Example 6-14.

Example 6-14 Required entries in the lldpad.conf file

```
t1vid001b3f00 :
{
    enableTx = true;
    fmode = "reflectiverelay";
    capabilities = "rte,ecp,vdp";
};
t1vid001b2101 :
{
    enableTx = true;
};
adminStatus = 3;
vdp :
{
    enableTx = true;
};
```

To verify the nearest bridge configuration for the network interface, use the command shown in Example 6-15.

Example 6-15 Verifying the nearest bridge configuration with lldptool

```
[root@kvm1 ~]# lldptool -i eth0 -t -V evbCfg -c enableTx -c fmode -c capabilities
enableTx=yes
fmode=reflectiverelay
capabilities=rte ecp vdp
```

To verify the nearest customer bridge configuration for the network interface, use the command shown in Example 6-16.

Example 6-16 Verifying the nearest customer bridge configuration with lldptool

```
[root@kvm1 ~]# lldptool -i eth0 -g ncb -t -V evbCfg -c enableTx -c fmode -c
capabilities
enableTx=yes
fmode=reflectiverelay
capabilities=rte ecp vdp
```

Updating the virt-manager package

Update the `virt-manager` package by using the `yum update virt-manager` command. If the update is not in your yum repository, use the `rpm update` command (Example 6-17).

Example 6-17 Updating virt-manager with the rpm update command

```
[root@kvm1 ~]# rpm -Uvh virt-manager-0.9.0-7.el6.x86_64.rpm
Preparing...          ##### [100%]
   1:virt-manager     ##### [100%]
```

You need to run virt-manager version 0.9.0-7 or higher.

Requirement: virt-manager-0.9.0-7 requires the following packages:

- ▶ spice-glib-0.6-2.e16.x86_64.rpm
- ▶ spice-gtk-0.6-2.e16.x86_64.rpm
- ▶ spice-gtk-python-0.6-2.e16.x86_64.rpm
- ▶ python-virtinst-0.600.0-5.e16.noarch.rpm

You know that you are running the correct version of the Virtual Machine Manager (virt-manager) if you can see the options shown in Figure 6-31. These options are displayed when you edit a virtual machine networking and select the **virtio** device model.

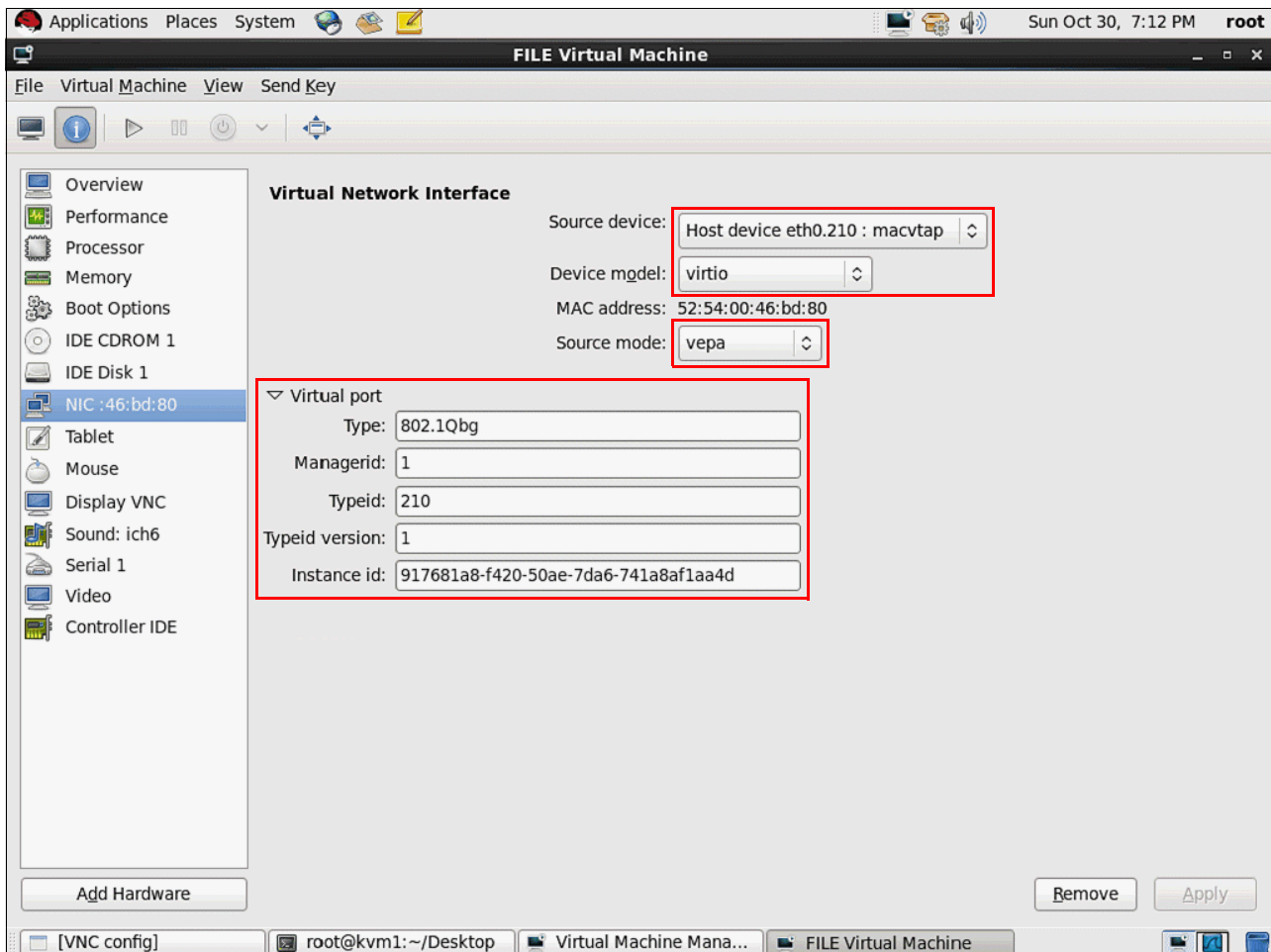


Figure 6-31 Virtual Machine Manager with the EVB networking options

Configuring the KVM domains network interfaces

As shown in Figure 6-31, EVB requires that you use a virtio device model. In addition, you must connect your virtual machine directly to a physical interface (or a VLAN interface) rather than use conventional bridges. This restriction means that you must install virtio device drivers in your virtual machines.

The procedure for installing virtio guest device drivers is available in the RedHat documentation or online at:

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/Chapter-Virtualization-KVM_Para_virtualized_Drivers.html

After you install the virtio guest network device drivers, you can modify your virtual machines virtual network adapter definition by using the Virtual Machine Manager:

1. Start Virtual Machine Manager and shut down the virtual machine you need to modify.
2. Open the virtual machine and go to the Details window.
3. Select the virtual machine network interface. The window shown in Figure 6-31 on page 176 is displayed.
4. In **Source device**, select the device that matches the VLAN you want the virtual machine to belong to. In this example, the VLAN is **Host device: eth0.210: macvtap**. The VLAN interface you select **must** match the VLAN ID assigned to the VSI type ID.

Tip: If you do not see the device you need, create the device configuration file in `/etc/sysconfig/network-scripts`. Example 6-18 shows an example network device configuration file.

Example 6-18 Sample network interfaces configuration files

```
[root@kvm1 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0.210
DEVICE=eth0.210
VLAN=yes
ONBOOT=yes

[root@kvm1 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT=yes
HWADDR=00:c0:dd:12:20:40
TYPE=Ethernet
BOOTPROTO=none
```

5. Select **virtio** in **Device model**.
6. Select **vepa** in **Source mode**.
7. Expand the Virtual port section, and enter 802.1Qbg in **Type**.
8. Enter the same manager id, type id and version id you created in the VSI database in Figure 6.3.2 on page 169. You do not need to enter the VSI instance id because it is automatically generated when you click **Apply**.

Requirement: The VLAN host interface selected must match the VLAN number specified in the VSI type ID definition in the VSI database. Otherwise your virtual machine is not authorized to associate with that VSI type ID and will fail to start.

Alternatively, you can configure your virtual machine XML definition by using `virsh edit <domain>`. Example 6-19 on page 178 shows a sample configuration (only the `<interface>` section is shown, which is in the `<devices>` section of the XML file). You do not need to enter the VSI instance id because it is added automatically when the virtual machine is first started with its new configuration.

Example 6-19 A sample virtual machine XML configuration with EVB networking

```
<interface type='direct'>
  <mac address='52:54:00:46:bd:80' />
  <source dev='eth0.210' mode='vepa' />
  <virtualport type='802.1Qbg'>
    <parameters managerid='1' typeid='210' typeidversion='1' />
  </virtualport>
  <model type='virtio' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
</interface>
```

You are now ready to start your virtual machine.

6.3.4 Verifying the EVB configuration

After you are done configuring all the components required for EVB, test your setup to verify that everything is working correctly.

Making sure that LLDP TLVs are exchanged

To verify that the physical switch and hypervisor are exchanging LLDP capabilities, on the switch, use the `/info/virt/evb/evb/local` and `/info/virt/evb/evb/remote` commands shown in Example 6-20. These commands check local capabilities (for the switch port) and remote capabilities (what has been negotiated with the hypervisor).

Example 6-20 Checking that LLDP TLVs are exchanged

```
>> VSI Information# /info/12/11dp/port 3/tlv/evb
Port Number: 3 (INT3)
      EVB TLV Status:      Local      Remote
      -----
Capability      STD RR RTE ECP VDP      RR RTE ECP VDP
Current Config  RR RTE ECP VDP      RR RTE ECP VDP
VSI Supported   256                    256
VSI Configured  254                    1
RTE             16                      16
```

The RR, RTE, ECP, and VDP capabilities are in the Current Config section for both local and remote because the switch and hypervisor are exchanging the correct set of capabilities.

Viewing the VSI associations

As explained in Chapter 2, “Introducing VMready” on page 11, when a virtual machine starts in an EVB enabled infrastructure, it is associated with the VSI id it requested. This association occurs only if it is authorized to do so, meaning that it is in the correct VLAN. To do so, the physical switch retrieves the VSI information from the VSI database. If the VSI database server is unavailable, it looks in its cached copy of the VSI database information.

With KVM, Ildpad enables that discovery process to happen for the virtual machines.

To see that process as it occurs, use the following procedure:

1. Open a telnet session on the physical switch or configure syslog with IBM System Networking Element Manager so you can see the VSI association messages as virtual machines are started.

2. Start a virtual machine by using either the Virtual Machine Manager or the `virsh start <domain>` command as shown in Example 6-21. Example 6-21 also shows that on the switch, an association message is logged.

Example 6-21 Starting a virtual machine with virsh and viewing the VSI association message

```
[root@kvm1 ~]# virsh start FILE
Domain FILE started

Oct 31  9:28:57 172.25.160.7 WARNING vm: VSI Type ID 210 Associated mac
52:54:00:46:bd:80 on port 3
```

3. Example 6-22 shows how you can view the VSI association maps on the physical switch.

Example 6-22 Showing the VSI maps on the switch

```
/info/virt/evb/vdp/vms
VSI Associations
TypeId  MAC                Vlan  Port  TxACL  RxEntry
-----  -
210     52:54:00:46:bd:80  210   3     256    50974
```

4. Migrate the virtual machine to another host as shown in Example 6-23. You should see both a dissociate message and a new association message for the virtual machine on another port.

Example 6-23 Viewing the dissociate and associate messages when a virtual machine migrates

```
[root@kvm1 ~]# virsh migrate FILE qemu+ssh://kvm2.local/system
Enter passphrase for key '/root/.ssh/id_dsa': *****

Oct 31  9:31:43 172.25.160.7 WARNING vm: VSI Type ID 210 Dissociated mac
52:54:00:46:bd:80 from port 3
Oct 31  9:32:03 172.25.160.7 WARNING vm: VSI Type ID 210 Associated mac
52:54:00:46:bd:80 on port 5
```

5. Example 6-24 shows that the VSI map is updated on the switch after the migration.

Example 6-24 The updated VSI map after a virtual machine migration

```
/info/virt/evb/vdp/vms
VSI Associations
TypeId  MAC                Vlan  Port  TxACL  RxEntry
-----  -
210     52:54:00:46:bd:80  210   5     256    50976
```

Note that you did not have to reconfigure any switch ports with a specific VLAN for the network to continue to operate for your virtual machine.

6.3.5 Troubleshooting EVB

This section addresses common EVB errors and their solutions as well as IBM NBOS commands that are useful in troubleshooting.

Common EVB errors and fixes

This section address some of the error messages you can get when EVB is not configured properly.

Error starting domain: error 4 during port-profile setlink on interface: Invalid argument

This error occurs when you try to start a virtual machine that is attached to a VLAN different from the VLAN ID specified in the VSI type (Figure 6-32).

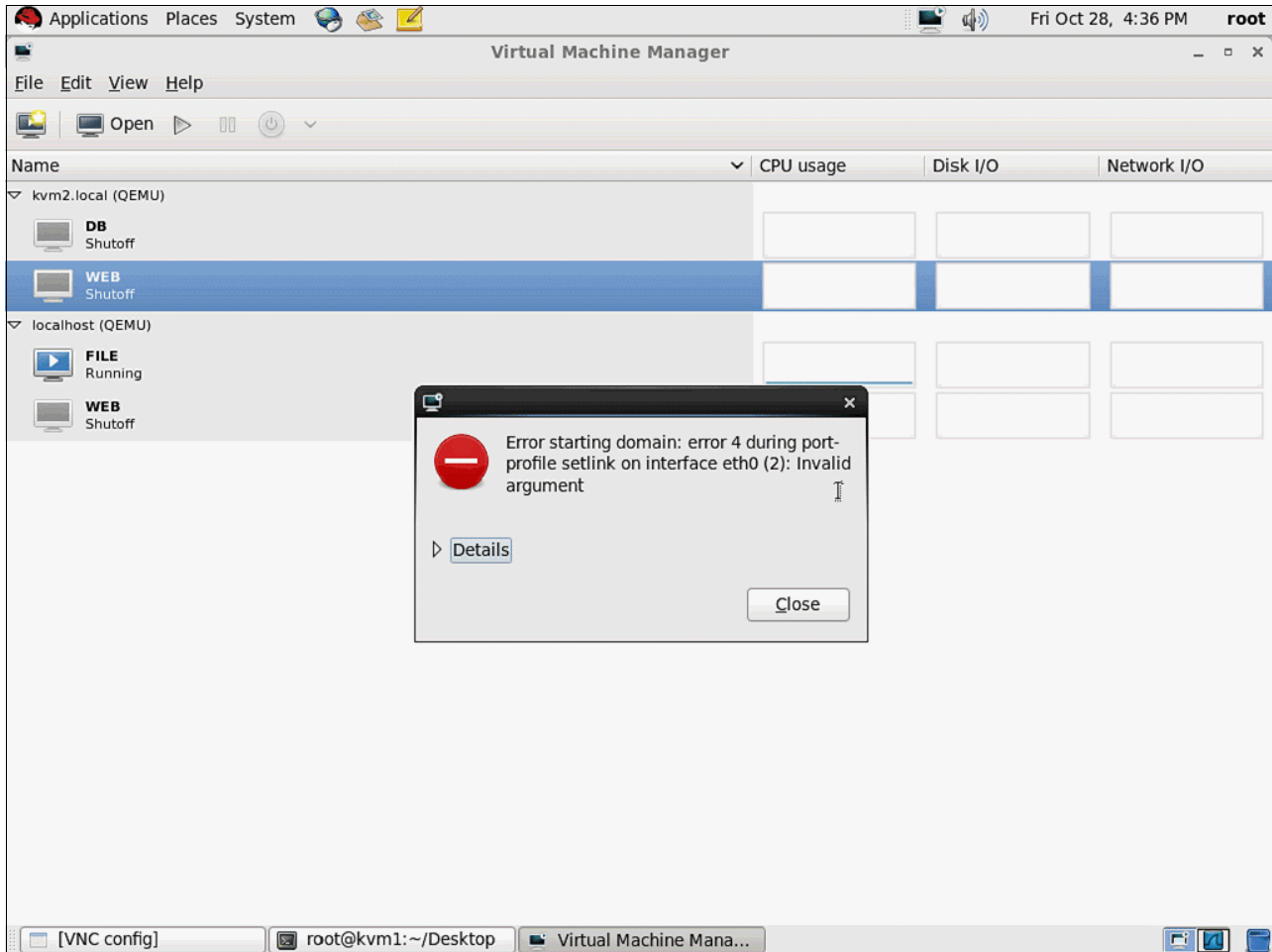


Figure 6-32 Error 4 when starting virtual machines

Example 6-25 shows the same error message when using `virsh` to start the virtual machine.

Example 6-25 Error 4 with virsh

```
[root@kvm1 ~]# virsh start FILE
error: Failed to start domain FILE
error: error 4 during port-profile setlink on interface eth0 (2): Invalid argument
```

Figure 6-33 on page 181 shows that the VM is attached to interface `eth0.200`. However, it is trying to associate with VSI type ID `210`, which in the setup has only VLAN 210 in its definition. Because those VLAN ids do not match, the association cannot happen.

Consideration: The VSI type ID does not need to match the virtual machine VLAN ID. However, the virtual machine VLAN ID must be assigned to the VSI type ID it is trying to associate with.

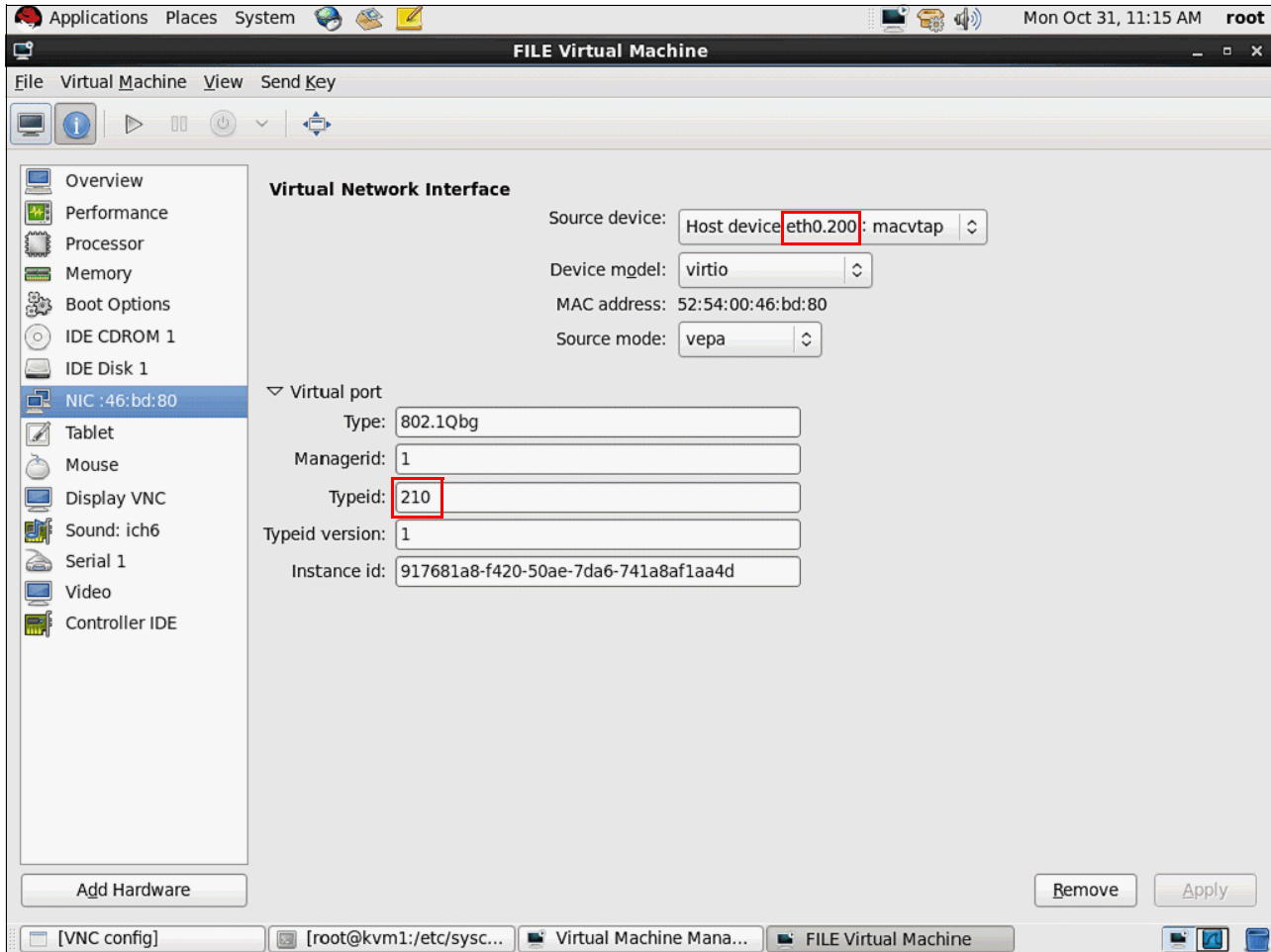


Figure 6-33 The incorrect virtual machine configuration that causes error 4

Select the appropriate VLAN interface for the virtual machine, and modify the VSI type ID or add the VLAN to the VSI type ID to correct this error.

This error can also happen if you specified a non-existent VSI type ID. It generates a message on the switch similar to the following: `WARNING vm: Error: VSI Type ID 220 doesn't exist`. In this case, either create the VSI type ID in the VSI database, or put the correct VSI type ID in the domain definition.

Error starting domain: internal error sending of PortProfileRequest failed

This error means that LLDP packets could not be sent. Also, the `lldpad` service is not running on the host where the virtual machine needs to be started or migrated to (Figure 6-34). Use the `service lldpad start` command to fix this error.

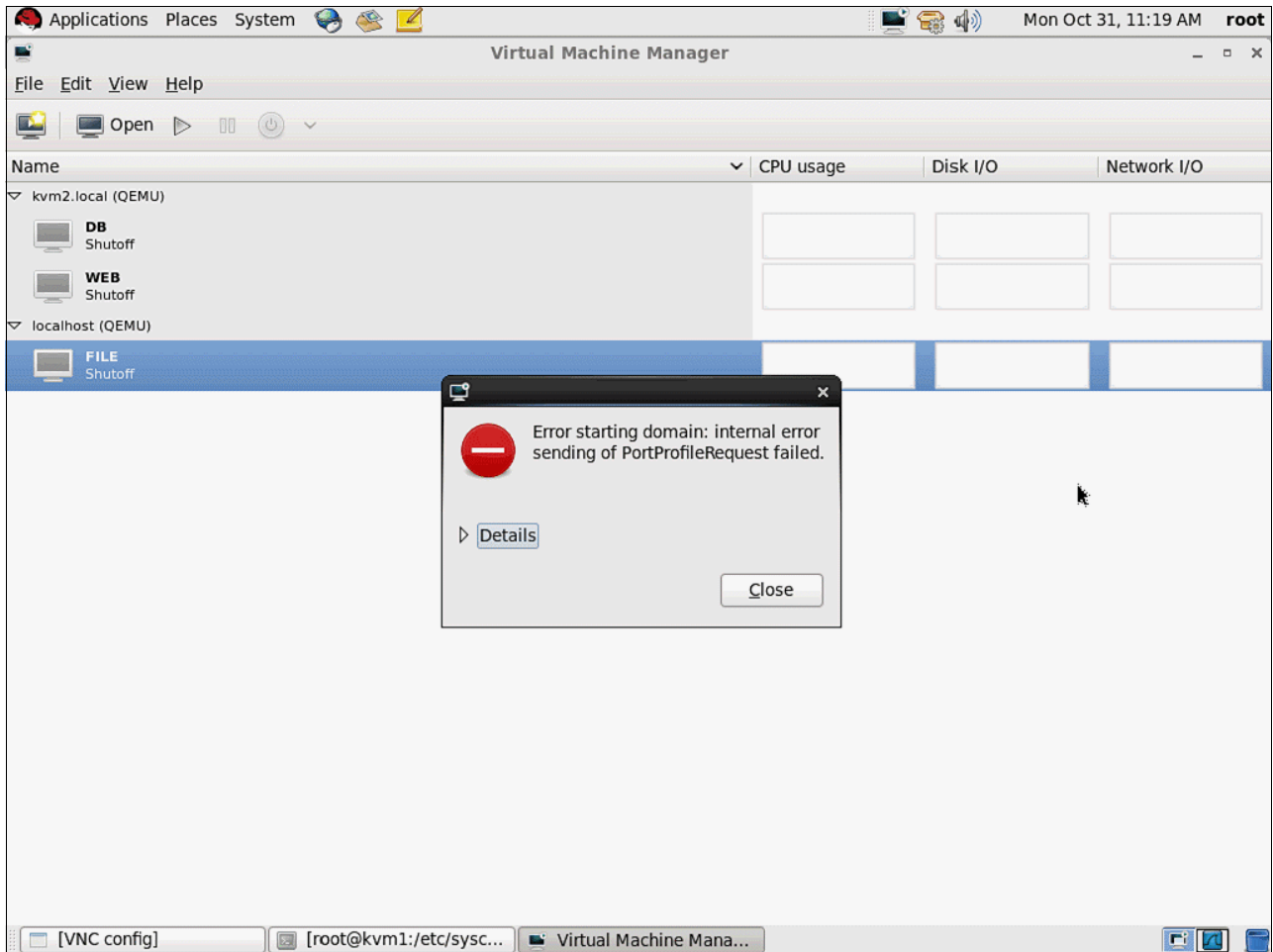


Figure 6-34 Error in Virtual Machine Manager when `lldpad` is not running

Example 6-26 shows the same error in `virsh`.

Example 6-26 Error in `virsh` when `lldpad` is not running

```
[root@kvm1 ~]# virsh start FILE
error: Failed to start domain FILE
error: internal error sending of PortProfileRequest failed.
```

Unable to migrate guest: internal error port-profile setlink timed out

This error, shown in Figure 6-35, can occur when `lldpad` is not working properly on the target host. You can also get this error when trying to start a virtual machine.

To fix it, restart `lldpad` with the `service lldpad restart` command.

Remember: As previously noted, the `lldpad` version used during testing was not fully functional, and had to be restarted manually to fix the VSI associations.

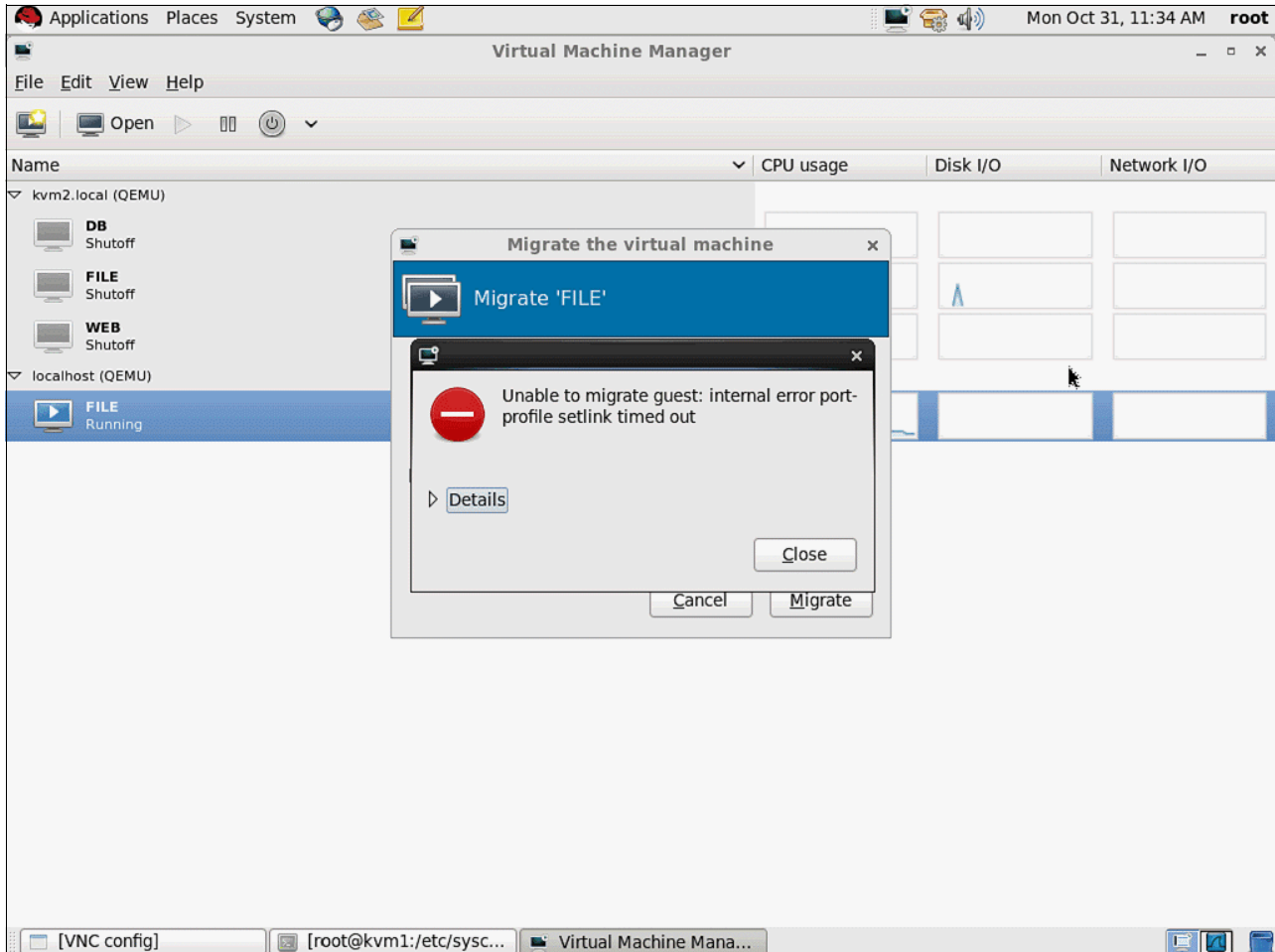


Figure 6-35 Error in virtual machine manager when `lldpad` is not working properly

Example 6-27 shows the same error occurring when using `virsh`.

Example 6-27 Error in `virsh` when `lldpad` is not working properly

```
[root@kvm1 ~]# virsh migrate FILE qemu+ssh://kvm2.local/system
Enter passphrase for key '/root/.ssh/id_dsa':
error: internal error port-profile setlink timed out
```

Useful IBM NOS commands

This section contains other information and techniques you can use when troubleshooting EVB. Table 6-1 lists useful commands that can be used when troubleshooting EVB from the switch side.

Table 6-1 Useful IBM NOS commands related to EVB

Command	Purpose
<code>/info/virt/evb/vdp/vsidb</code>	Displays the VSI database information that is currently in the switch cache.
<code>/oper/virt/evb/dbupdate</code>	Retrieves the VSI database information from the VSI database server.
<code>/oper/virt/evb/dbclean</code>	Clears the VSI database information cache in the switch.
<code>/cfg/virt/evb/vsidb/cur</code>	Displays the VSI database server configuration information in the switch.
<code>/info/virt/evb/vdp/vms</code>	Displays the current VSI associations on the switch.
<code>/oper/virt/evb/cleanvms</code>	Clears all current VSI associations on the switch.
<code>/info/12/11dp/port <port#>/tlv/evb</code>	Displays the local and remote supported and current LLDP capabilities that were negotiated between the switch and the hypervisor.

EVB VSI discovery error messages

Table 6-2 documents the VSI association error codes and what they mean.

Table 6-2 Summary of error codes for VSI associations

Error Number	Error Short Name	Error Text
1	Invalid Format	The VSI format is not supported by the switch.
2	Insufficient Resources	The switch does not have enough resources to complete the VSI operation successfully.
3	Unused VTID	The VSI referenced by the VSIID does not exist in the VSI Manager database referenced by the VSI Manager Identifier.
4	VTID Violation	The VSI referenced by the VSIID is not allowed to be associated with the VTID.
5	VTID Version Violation	The VSI referenced by the VSIID is not allowed to be associated with the VTID Version.
6	Out of Sync	The VTID or one of the VSI List fields used in the Associate is not the same as the corresponding field used in the Pre-Associate.

Running lldpad in debug mode

When you suspect the problem is with lldpad, restart the service in debug mode and redirect the output to a text file (Example 6-28). This process allows you to view all the LLDP messages that lldpad sends on behalf of the virtual machines to the physical switch.

Example 6-28 Starting lldpad in debug mode

```
[root@kvm1 ~]# lldpad -V 7 >/tmp/lldpad_debug.txt &
[1] 17277

[root@kvm1 ~]# service lldpad status
Checking for service lldpad: running

[root@kvm1 ~]# ls -l /tmp/lldpad*
-rw-r--r-- 1 root root 24576 Oct 31 14:00 /tmp/lldpad_debug.txt

[root@kvm1 ~]# virsh start FILE
Domain FILE started

[root@kvm1 ~]# virsh shutdown FILE
Domain FILE is being shutdown

[root@kvm1 ~]# service lldpad stop
Shutting down lldpad: [done] [ OK ]
[1]+ Done lldpad -V 7 > /tmp/lldpad_debug.txt
```

Example 6-29 shows a sample lldpad debug message.

Example 6-29 Sample lldpad debug message

```
vdp_add_profile(1312): adding vdp profile for eth0 !
profile 0x19880d0:

mode: 2 (VDP_MODE_ASSOCIATED)
response: 0 (success)
state: 0 (VSI_UNASSOCIATED)
mgrid: 1
id: 210 (0xd2)
version: 1
instance: 917681a8-f420-50ae-7da6-741a8af1aa4d
format: 0x0
entries: 1
mac: 52:54:00:46:BD:80
vlan: 210
```

Manually associating KVM domains with VSI type ids using lldptool

If a KVM virtual machine is no longer associated on the switch and has lost network connectivity, use lldptool to do a manual association. The lldptool is part of the lldpad package.

The syntax for lldptool manual association (as of lldpad-0.9.43-10.el6.x86_64) is:

```
lldptool -T -i ethx -V vdp -c mode=2,mgrid,typeid,typeidversion,instanceid,2,mac,vlan
```

You can then use `lldptool -t -i ethx -V vdp -c mode` to view the association messages.

Example 6-30 shows an example of both commands.

Example 6-30 Performing a manual association with lldptool

```
[root@kvm2 ~]# lldptool -T -i eth0 -V vdp -c  
mode=2,1,200,1,29fcff78-d3fd-4d1f-5456-381414c7a332,2,52:54:00:41:42:77,200
```

```
[root@kvm2 ~]# lldptool -t -i eth0 -V vdp -c mode  
mode=  
mode: 2 (VDP_MODE_ASSOCIATED)  
response: 0 (success)  
state: 2 (VSI_ASSOCIATED)  
mgrid: 1  
id: 200 (0xc8)  
version: 1  
instance: 29fcff78-d3fd-4d1f-5456-381414c7a332  
format: 0x2  
entries: 1  
mac: 52:54:00:41:42:77  
vlan: 200
```

You can get the VSI information (VLAN and MAC addresses) from the KVM domain XML file by using the `virsh edit <domain>` command. You can also get it from the Virtual Machine Manager as shown in Figure 6-36.

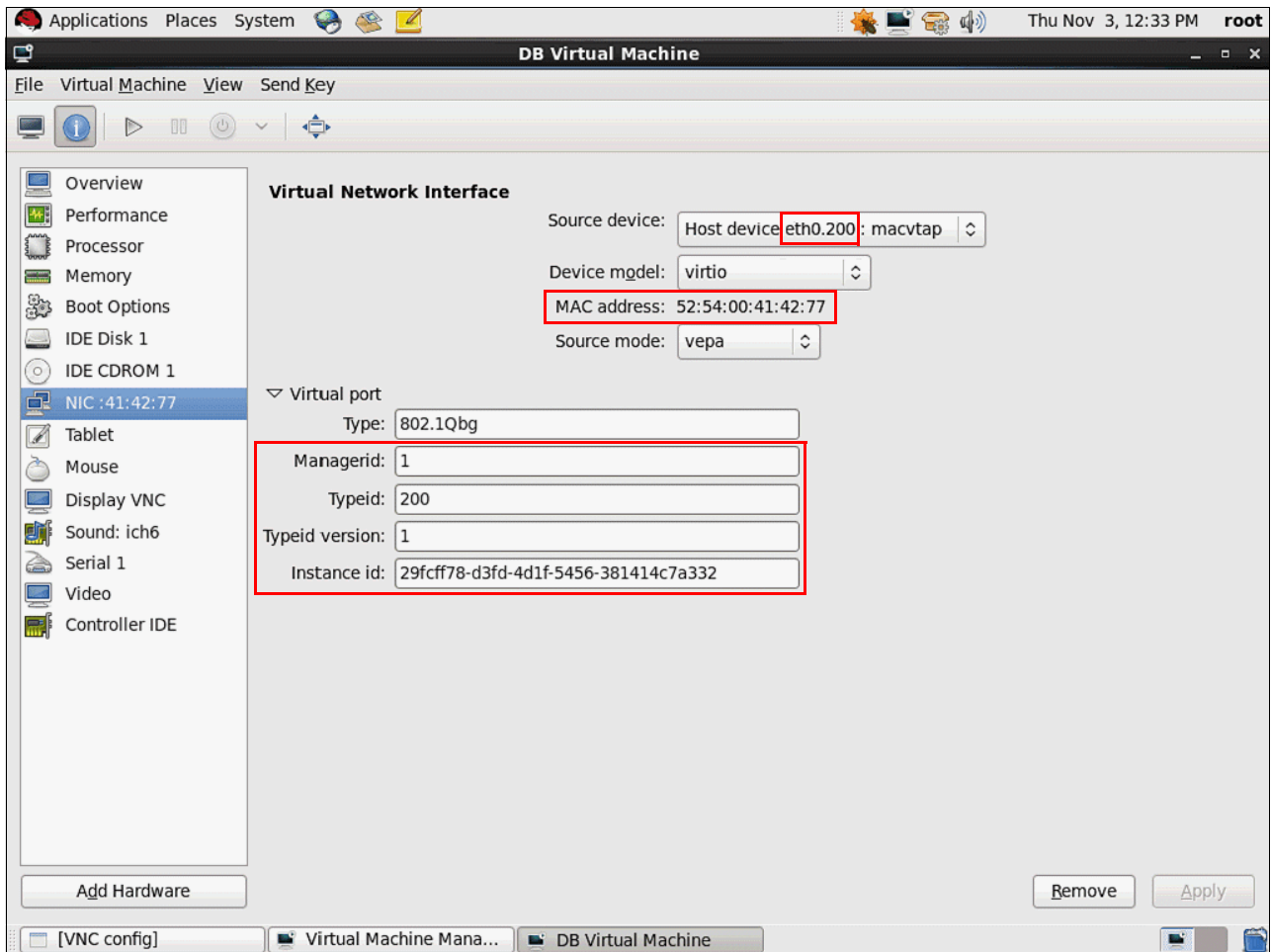


Figure 6-36 Viewing the VSI information and MAC address in Virtual Machine Manager

Viewing the physical switch syslog messages history

If you were not connected to the switch when errors occurred, you can view the syslog messages history by using the `/info/syslog ibmnos` command. You can also view the history from IBM System Networking Element Manager, as shown in Figure 6-37, if you configured IBM System Networking Element Manager to capture syslog messages from the switch.

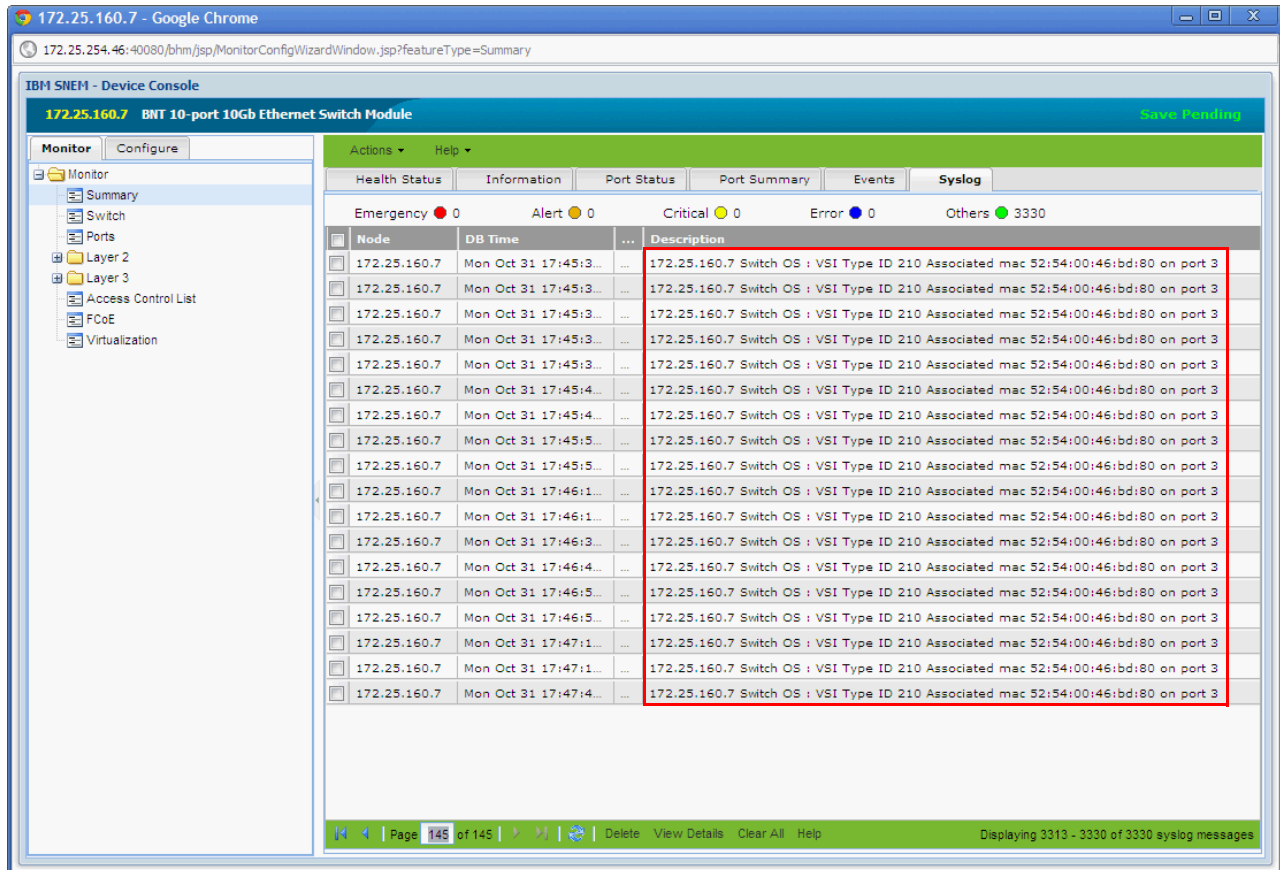


Figure 6-37 Viewing the syslog messages from IBM System Networking Element Manager 6.1

Tip: When IBM System Networking Element Manager 6.1 becomes generally available, the syslog facility can be moved to another component of the IBM System Networking Element Manager virtual appliance.

Capturing LLDP network packets using Wireshark

Updated versions of Wireshark enable LLDP packets parsing so that you can capture LLDP traffic. The following patches were used in the example setup to generate the data shown in Figure 6-38 on page 189:

- ▶ wireshark-gnome-1.2.15-1.el6.1.x86_64
- ▶ wireshark-1.2.15-1.el6.1.x86_64

Those packets are sent over the eth0 interface in the setup, which is the network interface used for virtual machine traffic.

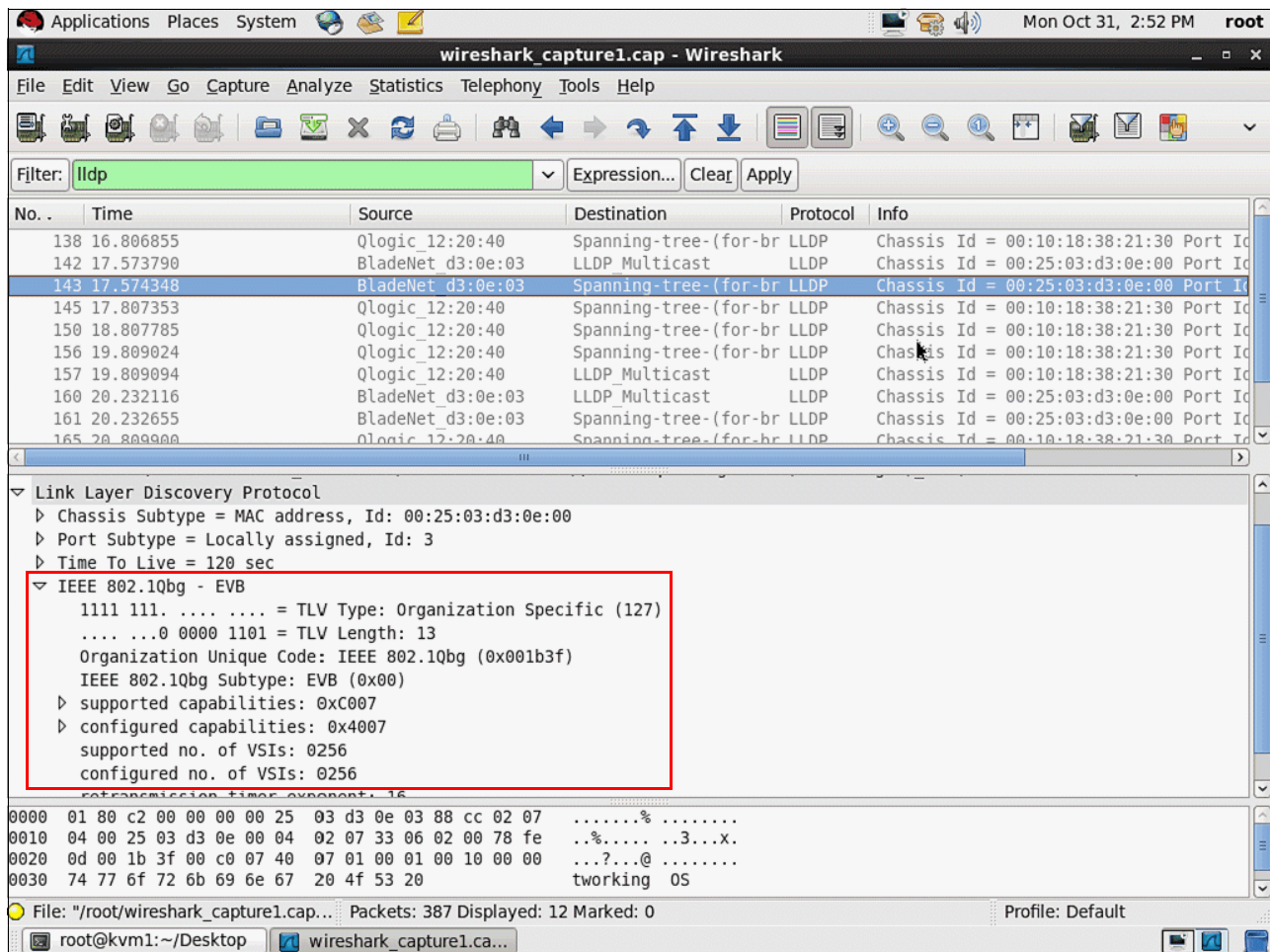


Figure 6-38 Viewing LLDP packets by using Wireshark

6.3.6 Implementing network policies with EVB

This section addresses how to implement virtual machine specific network policies in an 802.1Qbg environment. Specifically, it looks at how to implement access control lists and traffic shaping policies.

It also shows that those network policies continue to be applied to the virtual machine even after it is migrated to another switch port. This process requires no additional configuration.

Using access control lists

With EVB, you can define ACLs in the VSI database and then assign them to a VSI type ID. However, this type of ACL applies only to traffic that originates from virtual machines associated with that VSI type ID.

To block all traffic that goes to TCP port 80 in a VLAN in an EVB environment, use a standard VMAP on the switch. This VMAP must be applied to the entire VLAN.

An ACL continues to be applied after the WEB virtual machine is migrated to another host without requiring additional configuration on the switch. In addition, the ACL is also applied when VM to VM traffic on the same host is used. This was not the case with the VMready implementation without EVB support.

Example 6-31 shows how to create the ACL on the switch and assign it to VLAN 200, which is the Finance VLAN where the WEB server is located.

Example 6-31 Creating a VMAP and assigning it to a VLAN

```
/cfg/acl/vmap 2/action deny  
/cfg/acl/vmap 2/tcpudp/dport 80 0xffff  
/cfg/12/vlan 200/vmap add 2  
apply
```

Figure 6-39 shows that after the ACL is applied, the client PC cannot communicate with the WEB virtual machine over TCP port 80.

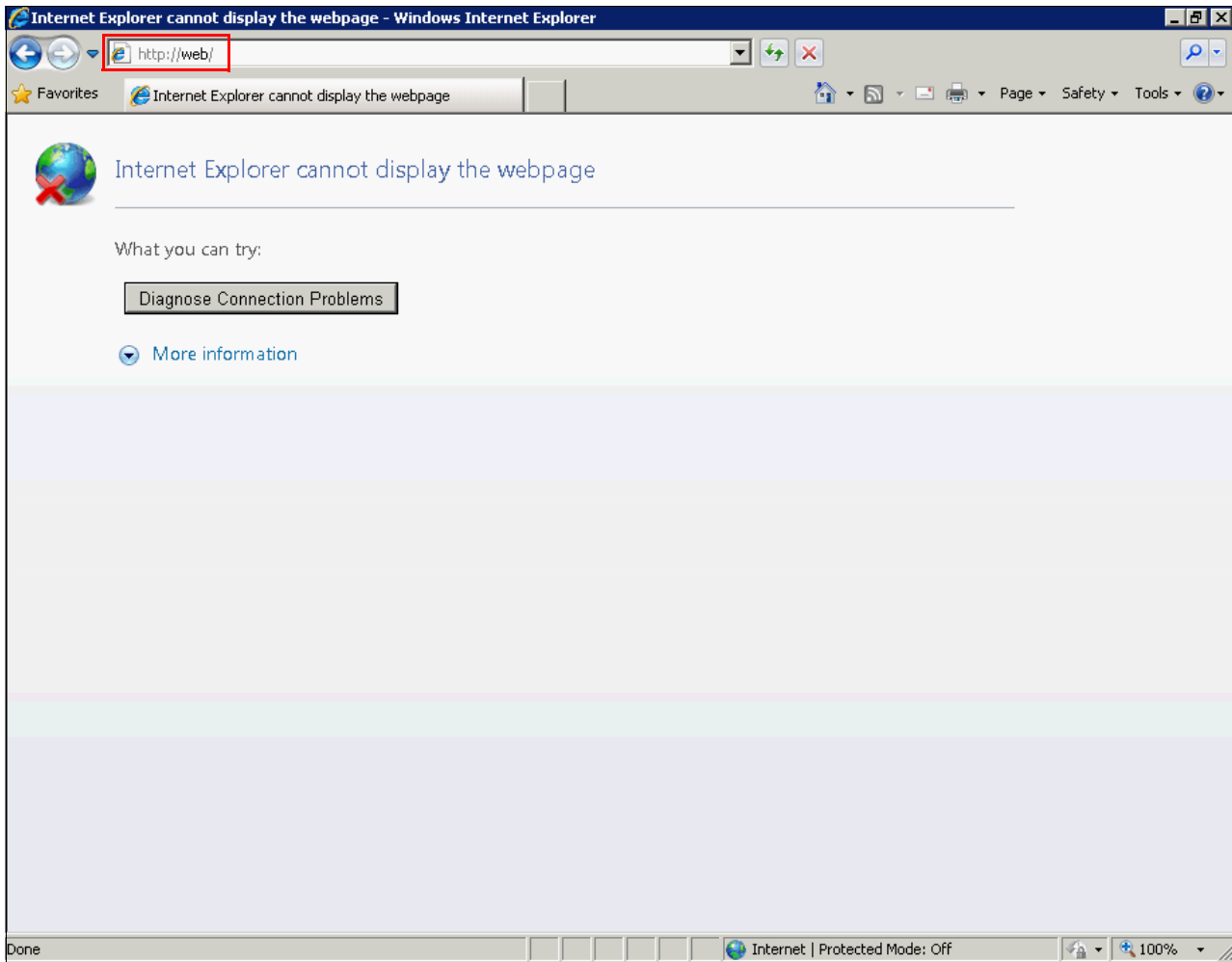


Figure 6-39 Showing that the client PC cannot connect to WEB over TCP port 80

Similarly, Figure 6-40 shows that after the ACL is applied, the DB virtual machine cannot communicate with the WEB virtual machine over TCP port 80.

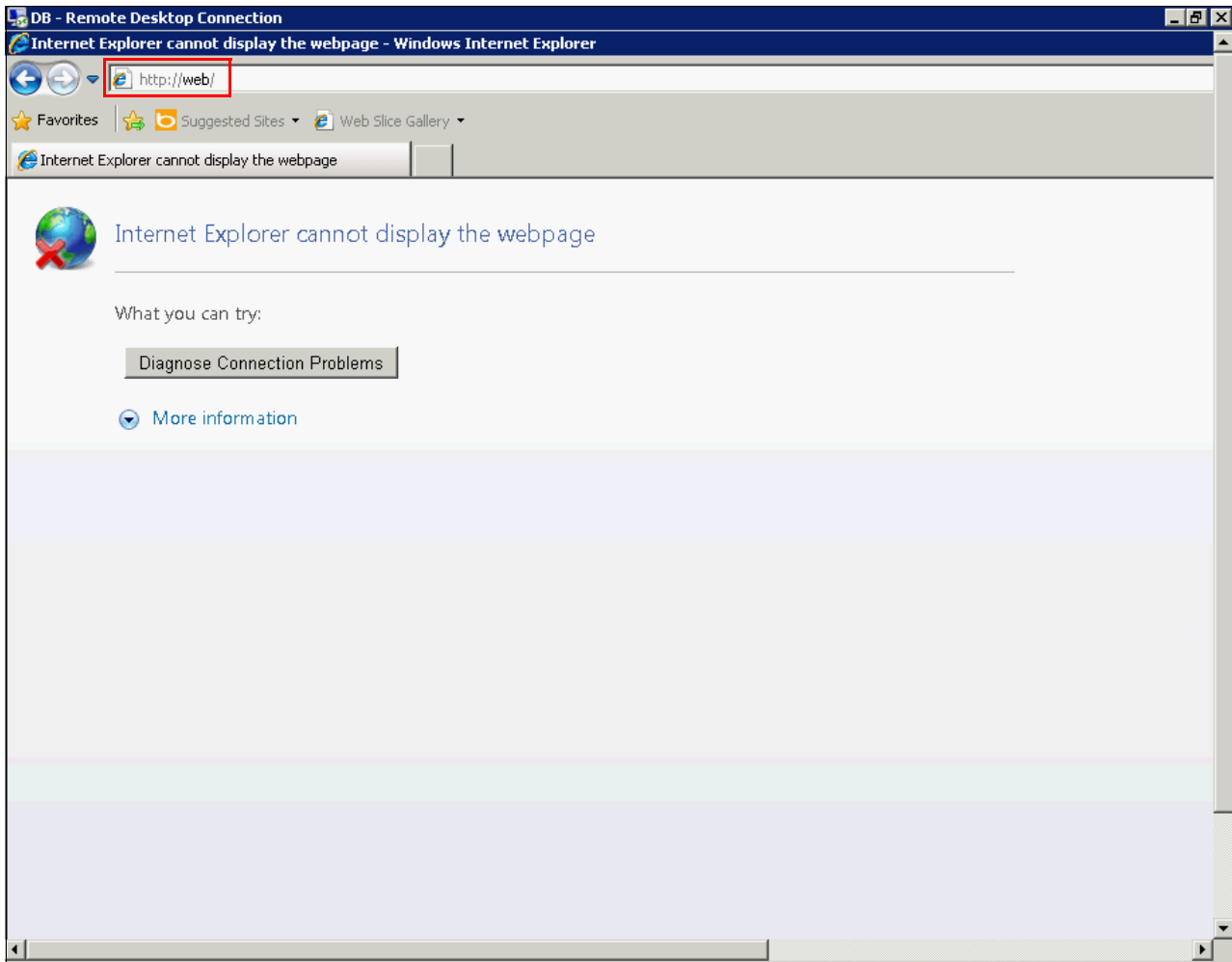


Figure 6-40 Showing that DB cannot connect to WEB over TCP port 80

The DB virtual machine is on the same host as the WEB virtual machine as shown in Figure 6-41.

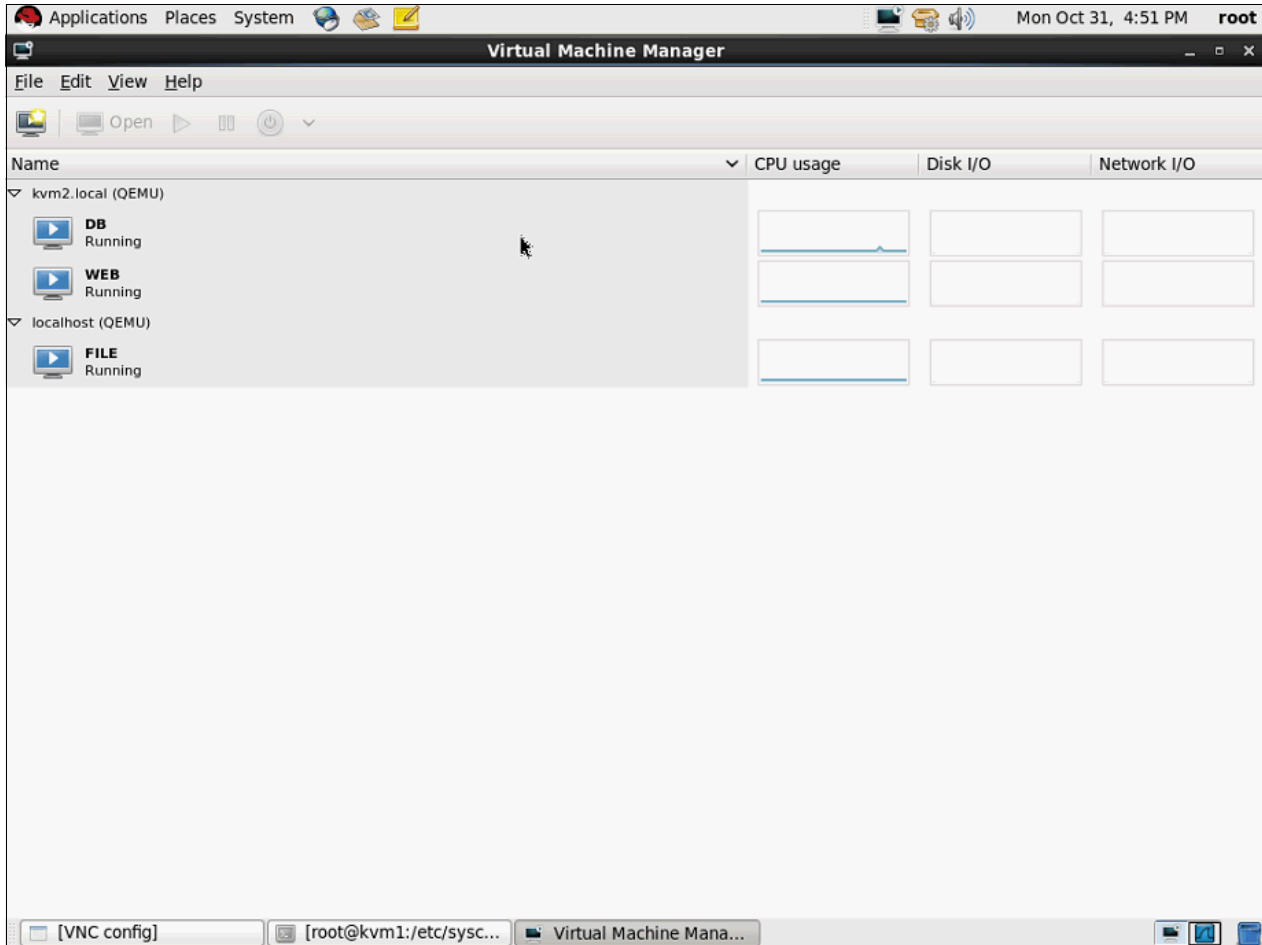


Figure 6-41 Showing that WEB and DB are running on the same host

If you migrate the WEB virtual machine to the other host, the ACL is still applied without requiring additional configuration.

Using traffic shaping policies

To apply traffic shaping by using EVB, modify the VSI type definition in your VSI database server. In this setup, this database is in the IBM System Networking Element Manager 6.1 server.

Figure 6-42 shows where the Tx and Rx settings are in the VSI type definition in the VSI DB console of IBM System Networking Element Manager 6.1.

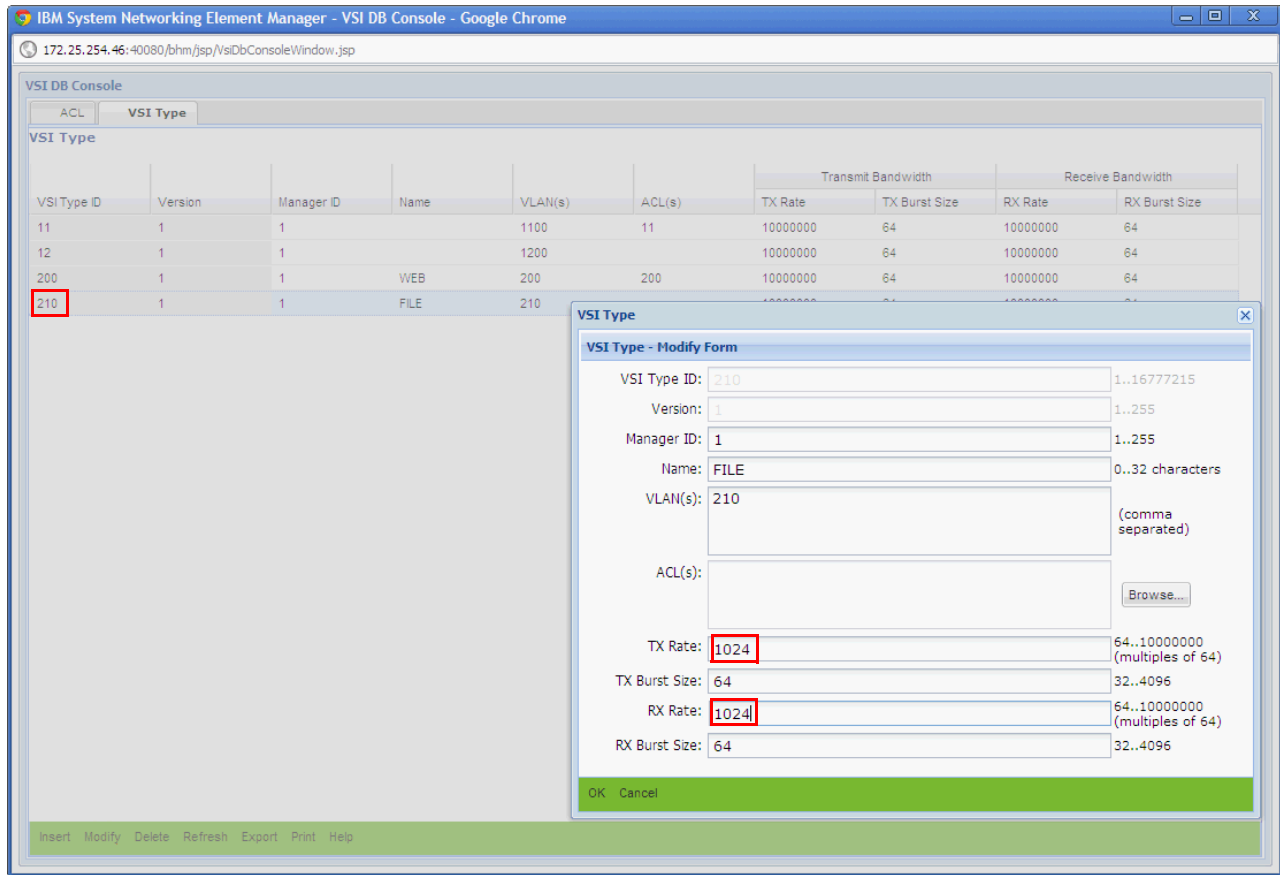


Figure 6-42 Configuring a traffic shaping policy on the VSI type ID in IBM System Networking Element Manager 6.1

Restriction: At the time of writing, Ildpad was not fully functional. Therefore, the traffic shaping policy changes were not picked up by the switch until an `/oper/virt/evb/dbupdate` was issued on the switch. For the policy to be applied, the virtual machine also had to be reassociated with the VSI type ID, either by migrating it to another host or restarting it.

Figure 6-43 shows the effect of the traffic shaping policy on the client PC.

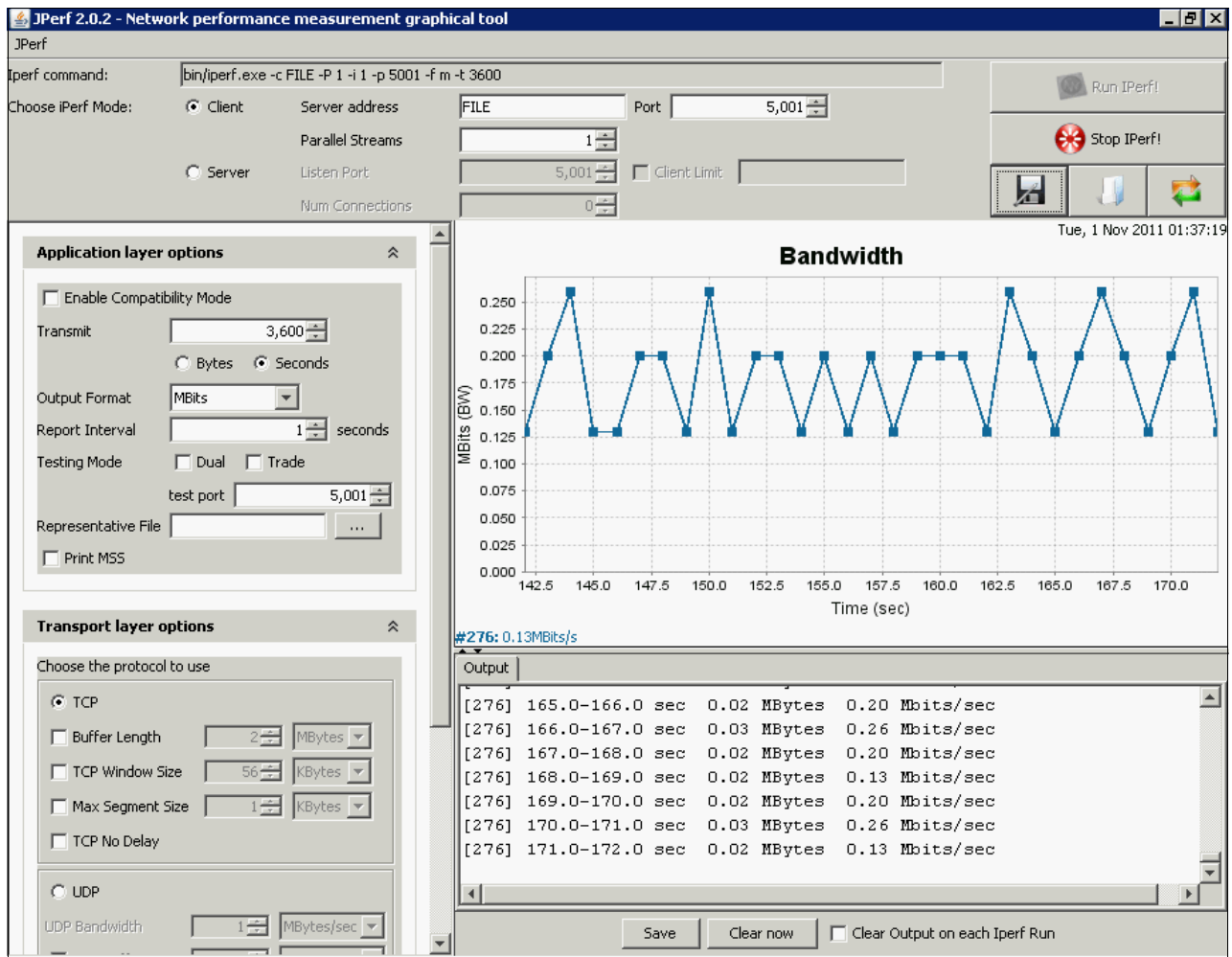


Figure 6-43 Showing the effect of the network traffic shaping policy on the client PC

If the virtual machine is migrated to another host, the policy will still be applied. This traffic shaping policy is also applied to VM to VM traffic on the same host as VEPA forces the network traffic to go to the physical switch.

6.3.7 Edge Virtual Bridging for KVM Frequently Asked Questions

This section contains a few frequently asked questions about EVB support on KVM that have not been addressed anywhere else in this chapter.

Do EVB and SR-IOV work together?

This configuration requires the embedded switch in the SR-IOV to support VEPA mode. At the time of writing, there were no adapters that implemented this configuration, but that function should be added in the next few months.

How does EVB work with network interface bonding?

EVB works with 802.3ad Link Aggregation because it appears as a single link to the operating system. At the time of writing, no other mode were supported because additional work was

required to associate over the passive link when the active link fails. Note that EVB with 802.3ad was not tested in the setup.

Will “classical” VMready and EVB work together?

As detailed in 6.3.1, “Step 1: Configuring the physical switch” on page 167, enabling EVB support on a VMready switch currently requires disabling “classical” VMready. Therefore, the two will not work together.

Integration of both VMready and VMready with EVB is planned for a future release of the IBM Networking Operating System.

Do I need high availability for my VSI server database component?

When the VSI database server is not available, the physical switch uses the local copy of the database in its cache and continues to service association requests.

Obviously, no additional VSI types can be created or associated until the VSI server database is available again. To compare it to another hypervisor, this impact is similar to what happens when you use distributed switches in VMware and the vCenter server becomes unavailable.

How do I provision new virtual machines in an EVB environment?

You can use the Virtual Machine Manager to provision new virtual machines in your EVB environment.

When you get to the last step of the virtual machine deployment wizard, make sure that you select **Customize configuration before install** as shown in Figure 6-44.

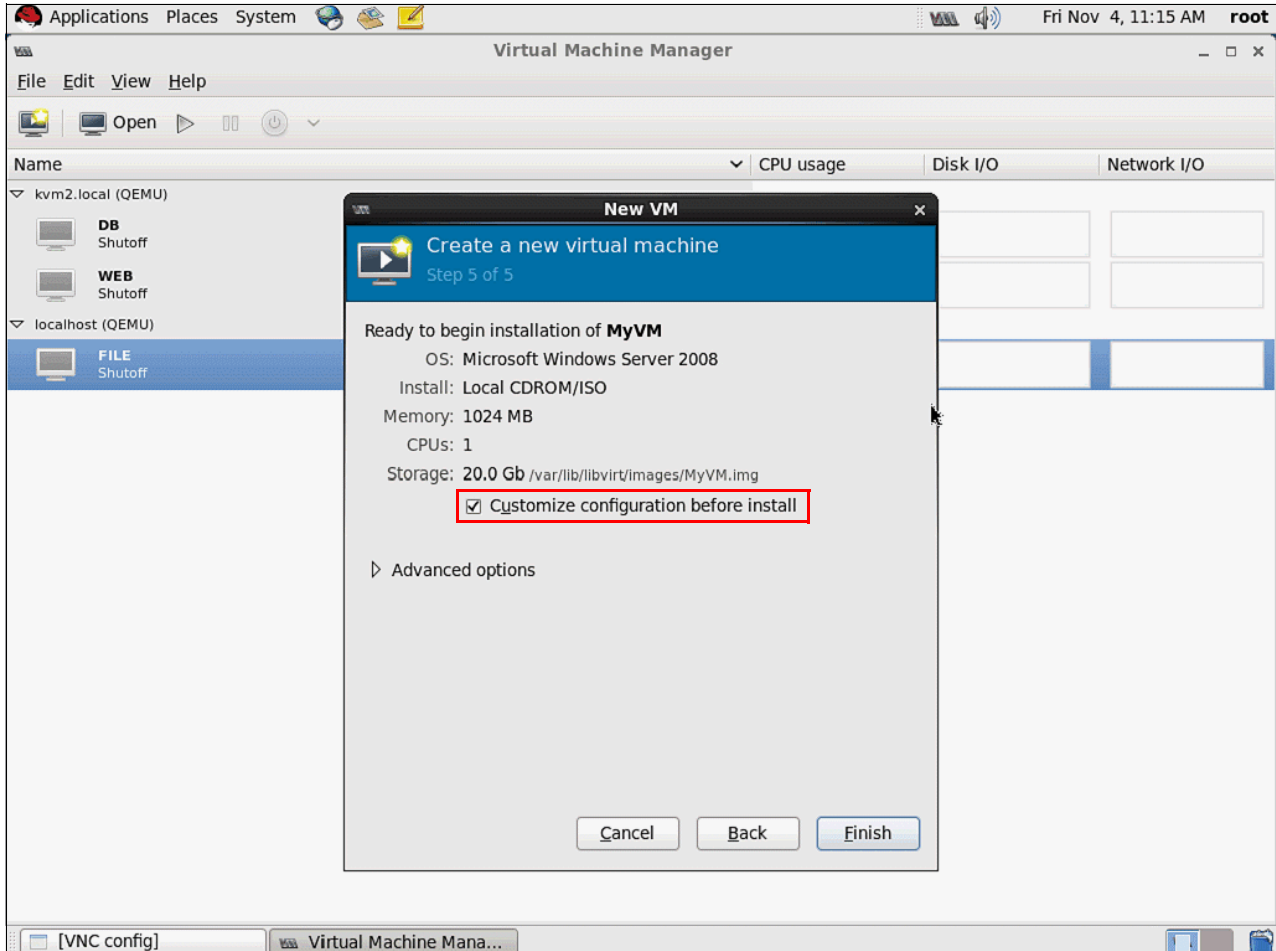


Figure 6-44 Customizing a domain configuration before installation in virt-manager

Click **Finish**. You can then enter the full NIC 802.1Qbg information as shown in Figure 6-45 before you continue with the installation. Make sure that you select **virtio** in the **Device model** field.

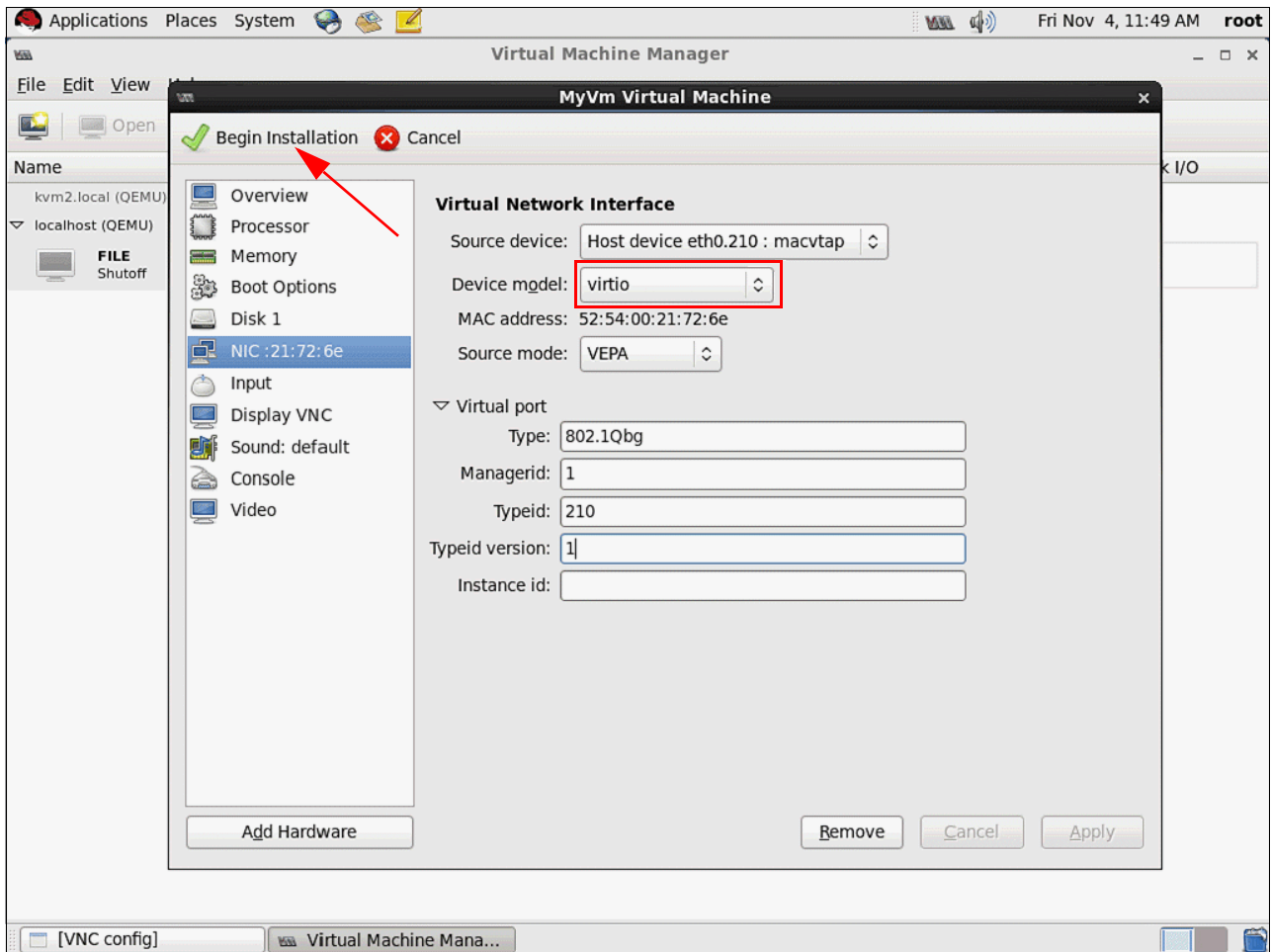


Figure 6-45 Entering the 802.1Qbg NIC information before starting the installation of a new domain in virt-manager

6.4 Conclusion

This chapter addresses how VMready can help automate network configuration in virtualized environments. It also shows how it enhances network control and security by making the network aware of virtual machines.

There are two possible solutions:

- ▶ VMready without 802.1Qbg or EVB support: This solution does not require any modifications in the hypervisor or in the virtual machines that it runs. It does require network administrators to validate each individual virtual machine that needs to come online in the network. These systems need to be added to the correct vmgroup.
- ▶ VMready with 802.1Qbg or EVB support: This solution requires firmware support on the physical switch and the introduction of a new component (IBM System Networking Element Manager). It also requires multiple configurations in the hypervisor and that the KVM virtual machines use virtio network device drivers. It enables the network administrator to use a central location for network policy configurations (the VSI database). These configurations are then published in the network and made available for virtual machines to subscribe to.

Both solutions have different requirements, but they address the same basic problems in virtual infrastructures. VMready with EVB enables additional automation and does not require network administrators to take action every time a new virtual machine needs to be provisioned.



Implementing VMready to support Hyper-V and other virtualization environments

This chapter describes the steps involved in implementing VMready for Hyper-V and other virtualization environments. This chapter includes the following sections:

- ▶ Overview of VMready for Hyper-V
- ▶ Initial configuration
- ▶ Initial tests
- ▶ Enabling VMready and implementing network policies
- ▶ Validating the network policies
- ▶ Validating Nmotion

7.1 Overview of VMready for Hyper-V

IBM VMready enables the network to automatically discover Hyper-V virtual machines and apply network settings to them. It also automatically moves network settings when virtual machines are moved.

7.1.1 A little about Hyper-V networking

Hyper-V is the second generation of Microsoft bare-metal hypervisor. Because it is still a young product, its network policy functions are limited to VLAN implementation. VMready can add some additional features such as traffic shaping to limit the bandwidth for a business group of servers. VMready can also add access control lists to improve the network security.

7.1.2 Example setup

As shown in Figure 7-1, the implementation and example scenario is based on two host systems, three virtual machines (File, Web, and DB), and a client system. The client system is used to demonstrate the impact of implementing network policies and show that those policies are preserved as virtual machines are migrated.

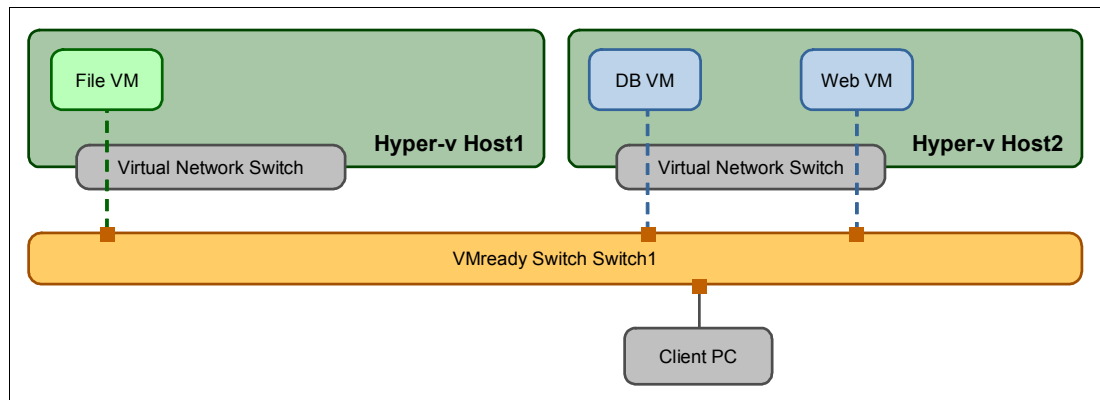


Figure 7-1 Example setup

7.1.3 The example scenario

To demonstrate VMready functions, this chapter uses the following scenario: A company provides several web applications to its employees. Network administrators want to enforce the security. They want to block all non-encrypted connections on port 80 on all the web servers and allow encrypted connections on port 443. In addition, A file server is available for the employees to download documents. Network administrators want to limit the bandwidth of this server.

Demonstrating this scenario involves these steps:

1. Describe the hardware and software environment of the scenario
2. Make some initial tests to highlight the weakness of the current environment
3. Enable VMready and implement network policies
4. Demonstrate the impact of these network policies
5. Migrate the virtual machines and show that their network settings automatically moved with them.

7.2 Initial configuration

This section addresses the initial segments of the configuration.

7.2.1 Hardware

The example environment is composed of one BladeCenter H. Two dedicated blade servers are positioned in bay #9 and bay #10 of the chassis as shown in Figure 7-2. The chassis has two IBM BNT 1/10 Gb Uplink Ethernet Switch Modules in bay #1 and bay #2. It also has two IBM BNT Virtual Fabric 10 Gb Switch Modules in bay #7 and in bay #9. Each blade has a QLogic 2-port 10 Gb Converged Network Adapter (CFFh).

The screenshot shows the IBM BladeCenter H Advanced Management Module interface. The main content area displays a table of blades. The table has columns for Bay, Status, Name, Pwr, Owner (KVM, MT), cKVM, I/O Compatibility, WOL, and Local Control (Pwr, KVM, MT). Blades 9 and 10 are highlighted with a red box.

Bay	Status	Name	Pwr	Owner**		cKVM*	I/O Compatibility	WOL*	Local Control			BEM
				KVM	MT*				Pwr	KVM	MT*	
1	On	Dev1	On				OK	On	✓	✓	✓	---
2	On	Dev2	On				OK	On	✓	✓	✓	---
3	Off	RHEL6-KVM1	Off	✓	✓		OK	On	✓	✓	✓	---
4	On	Dev3	On				OK	On	✓	✓	✓	---
5	On	RHEL6-KVM2	On				OK	On	✓	✓	✓	---
6	On	ESX-INFRA	On				OK	On	✓	✓	✓	---
7	On	ESX1	On				OK	On	✓	✓	✓	---
8	On	ESX2	On				OK	On	✓	✓	✓	---
9	On	HYPER-V1	On				OK	On	✓	✓	✓	---
10	On	HYPER-V2	On				OK	On	✓	✓	✓	---
11	Off	Free	Off				OK	On	✓	✓	✓	---
12	On	POWER2	On				OK	N/A	✓	✓	✓	---
13	On	RHEL5-XEN1	On				OK	On	✓	✓	✓	---
14	On	RHEL5-XEN2	On				OK	On	✓	✓	✓	---

* MT = Media Tray (CD/ USB) , WOL = Wake on LAN , BEM = Blade Expansion Module
 BSE1 (BSE2,BSE3) = Blade Storage Expansion 1st Generation (2nd Generation, 3rd Generation)
 PEU1 = PCI Expansion Unit 1st Generation PEU2 = PCI Expansion Unit II BPE3 = PCI Express Expansion Unit
 cKVM = Concurrent KVM Expansion BIE = Blade I/O Expansion BPR = Blade Processor Expansion

** You can change the KVM and Media Tray ownership on the Remote Control panel (under Blade Tasks).

Figure 7-2 BladeCenter overview

7.2.2 Hypervisor

Install Windows 2008 R2 Enterprise Edition SP1 on the two dedicated blade servers, and enable Hyper-V on these servers. Create a failover cluster composed of the two Hyper-V hosts as shown in Figure 7-3. These hosts are named HyperV1 and HyperV2.

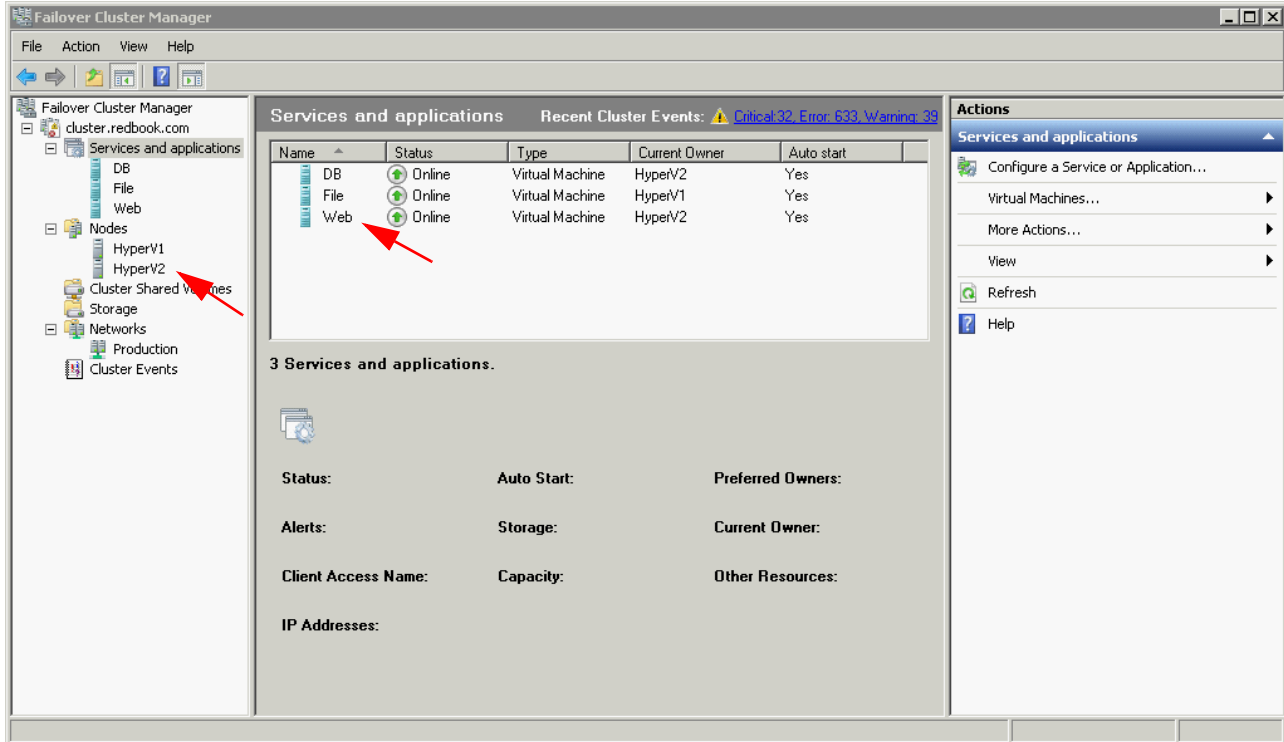


Figure 7-3 Failover cluster overview

7.2.3 Shared storage

The Hyper-V hosts have two shared LUNs on an iSCSI appliance. The first LUN is configured as a Cluster Shared Storage to host the virtual machine disks. The second LUN is used for the failover cluster Quorum configuration.

7.2.4 Network

The example configuration uses the first and the second IBM BNT 1/10 Gb Switch Modules in bay #1 and bay #2. These modules support the hypervisor management traffic such as the parent partition of the Hyper-V hosts and the iSCSI traffic. It uses Broadcom Advanced Control Suite 3 to create a NIC teaming of the two cards. In addition, a failover link is set up as shown in Figure 7-4.

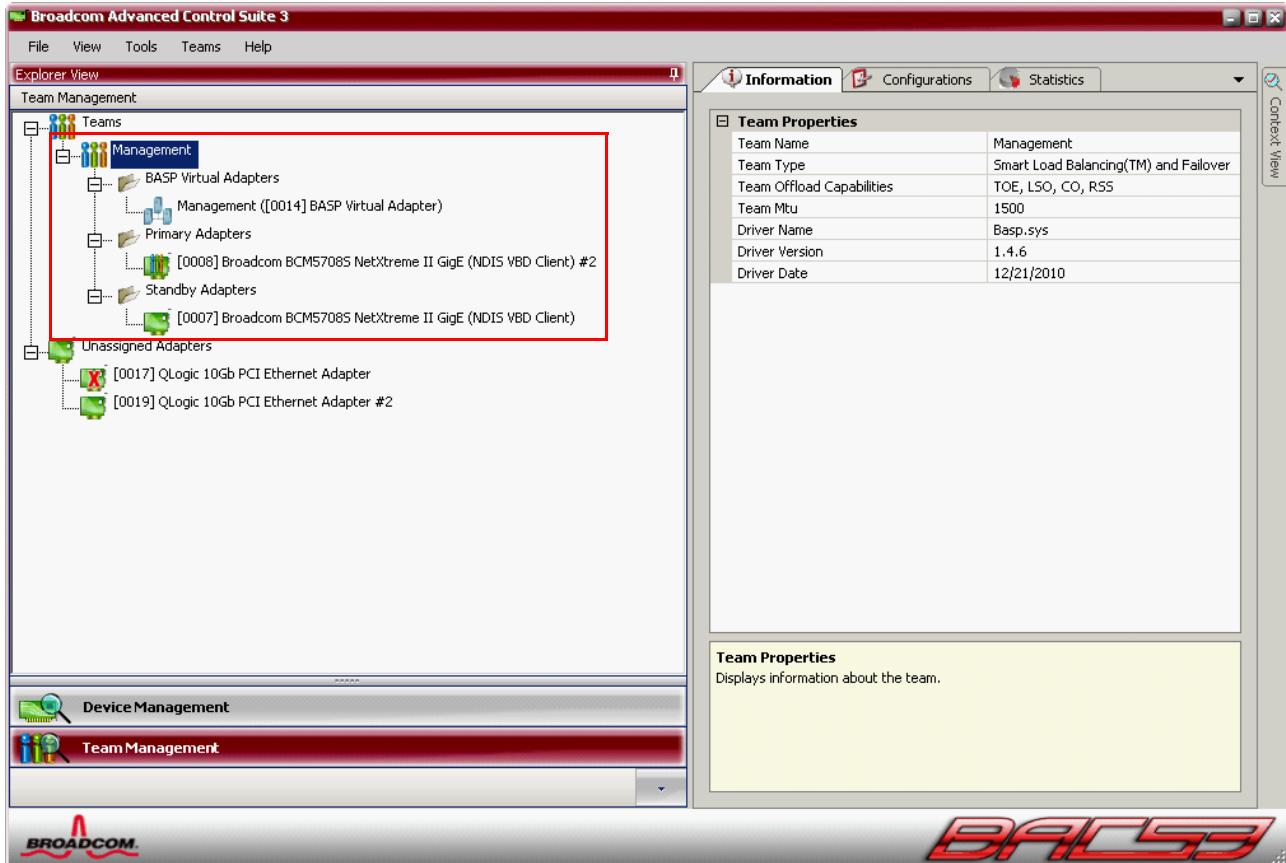


Figure 7-4 Network teaming configuration for the Hyper-V management traffic

The IBM BNT 10 Gb Virtual Fabric Switch Module in bay #9 is used to support the virtual machine network traffic. Configure this NIC in the Virtual Network Manager as an **External** connection type. Name that virtual network **Production** on HyperV1 and on HyperV2, as shown in Figure 7-5.

Restriction: At the time of writing, VMready does not support the virtual machine network traffic that goes through a NIC teaming configured with the Broadcom Advanced Control Suite. The suite prevents the virtual machines from being discovered on the IBM BNT switches.

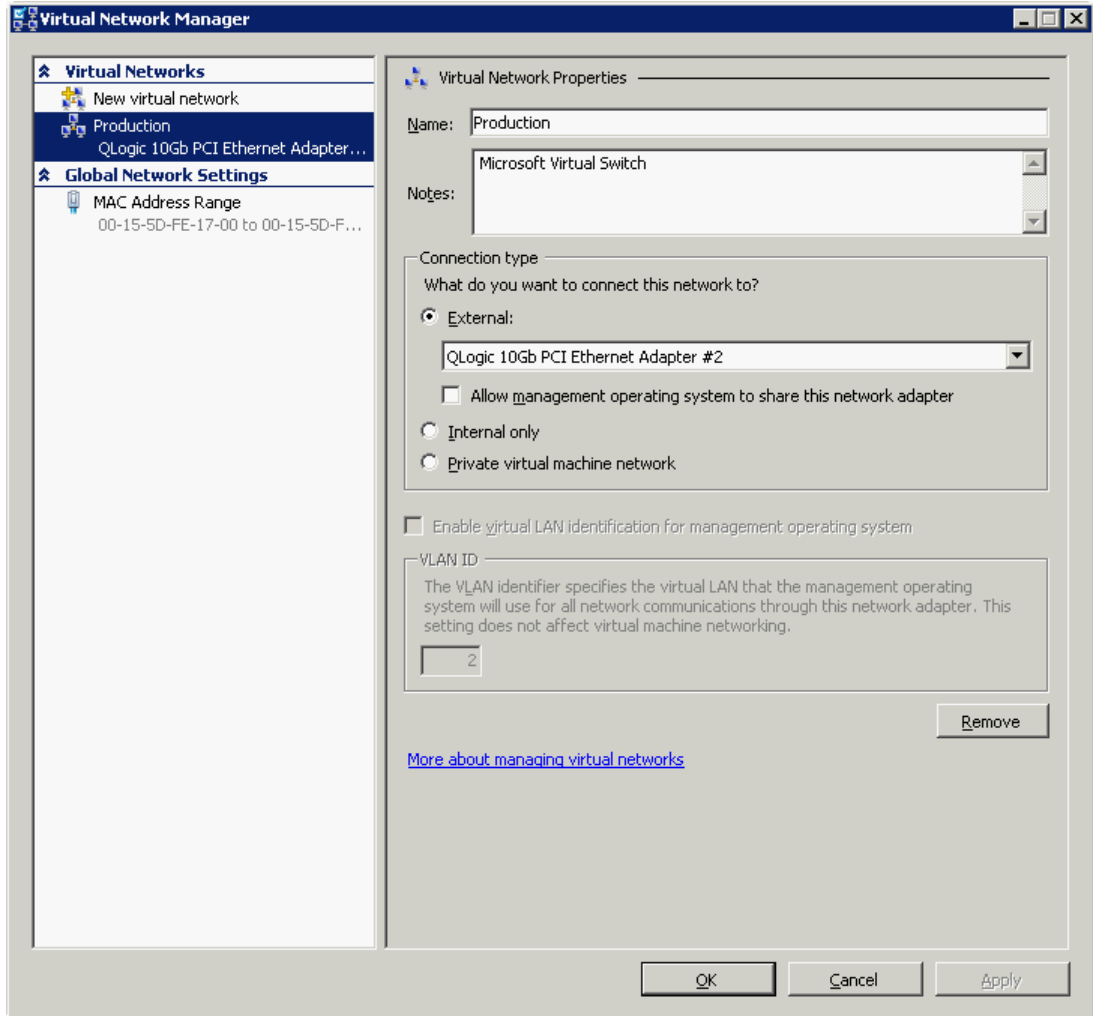


Figure 7-5 Production network in the Virtual Network Manager settings

7.2.5 Virtual machines

Provision three virtual machines named File, DB, and Web. These three VMs are three Windows 2008 R2 servers. Web server is configured with Microsoft Internet Information Services (IIS). It listens on port 80 for non-secure HTTP connections and on port 443 for secure HTTPS connections. Web and DB servers are on the same VLAN #300. File virtual machine is configured on a separate VLAN #400.

HyperV1 hosts the File virtual machine. HyperV2 hosts the DB and Web virtual machines. The virtual machine disks are hosted on the Cluster Shared Storage. Connect the virtual machines on the virtual network Production. VLAN tagging is enabled on the virtual machine ports as shown in Figure 7-6.

The virtual machines IP addresses have the following configuration:

- ▶ Web server: 192.168.3.101 / 255.255.255.0 (VLAN #300)
- ▶ DB server: 192.168.3.102 / 255.255.255.0 (VLAN #300)
- ▶ File server: 192.168.4.200 / 255.255.255.0 (VLAN #400)

The following scenarios use a Windows client to simulate the employee workstations. It is not hosted on those Hyper-V systems. For the demonstration, this client has two network interfaces: One on VLAN #300 and the other one on VLAN #400.

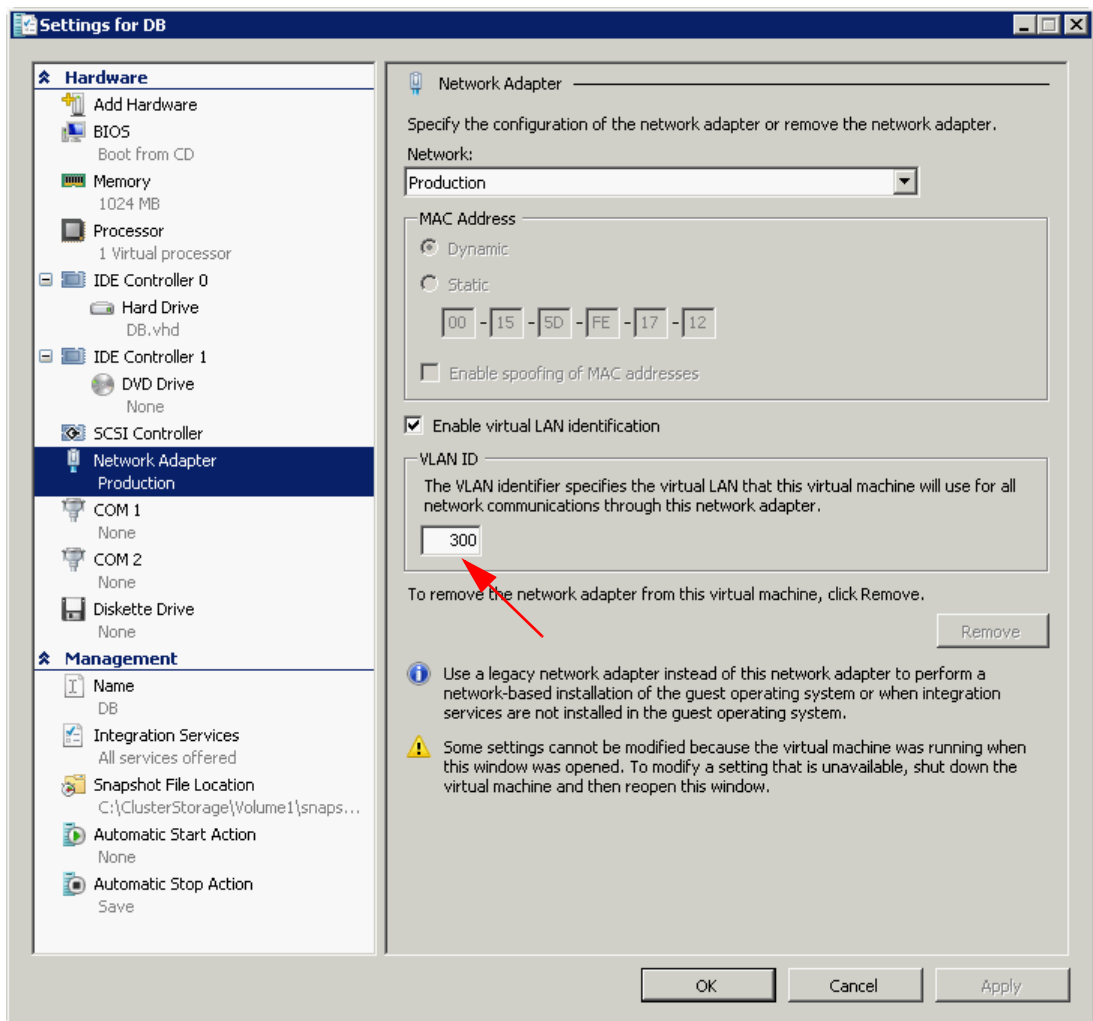


Figure 7-6 Initial network configuration for DB virtual machine

Figure 7-7 shows the initial configuration for the File virtual machine.

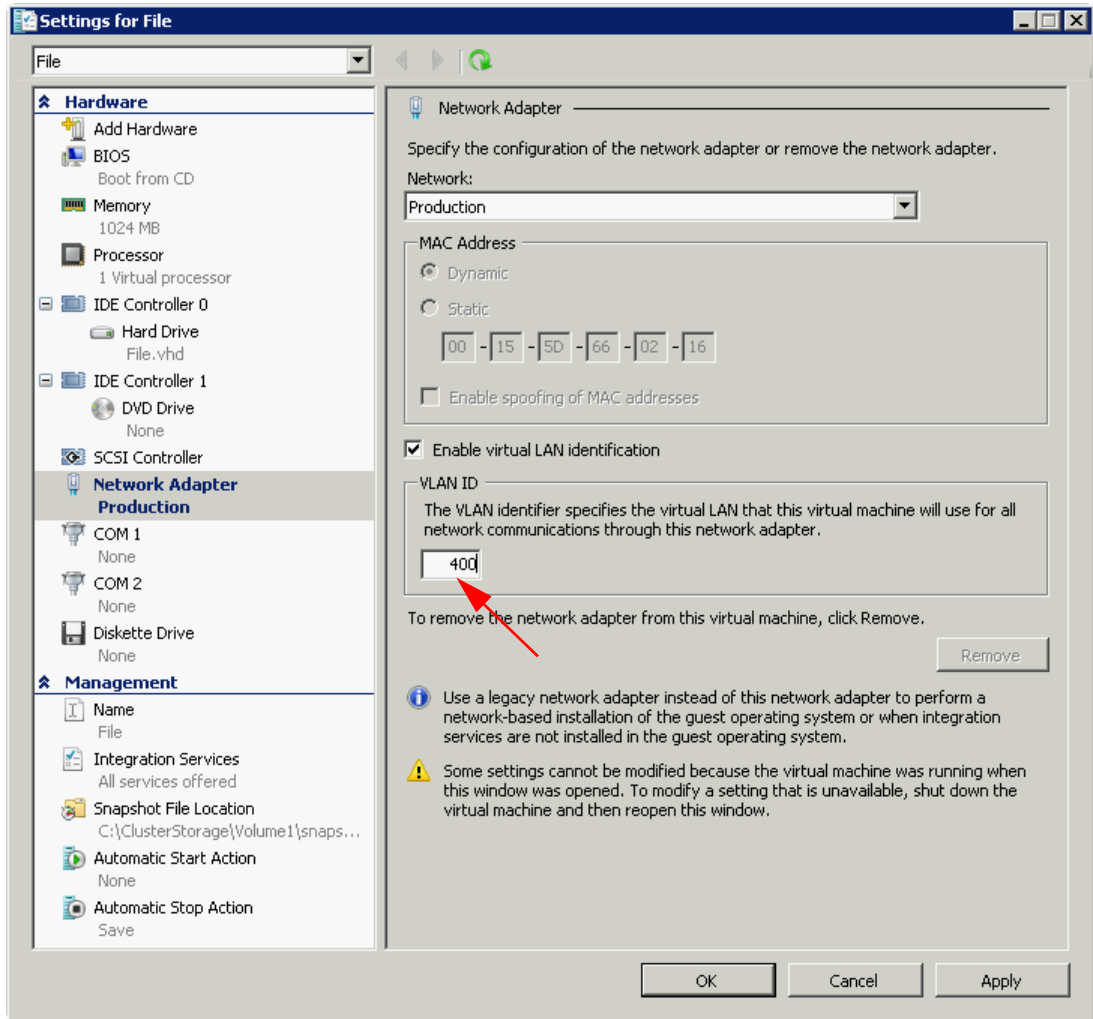


Figure 7-7 Initial network configuration for File virtual machine

7.2.6 Switch configuration

Figure 7-8 shows the current VLAN configuration on IBM BNT Virtual Fabric 10 Gb. Both ports INT9 and INT10 of the switch are configured with these settings:

- ▶ With PVID set to VLAN #2 and PVID tagging enabled (all the packets that are not tagged are automatically tagged with this garbage VLAN ID),
- ▶ As members of VLAN #300 and #400 (all the packets tagged with VLAN #300 or #400 are transmitted in the network).

```
Router#sho run
Current configuration:
!
interface port INT9
    tag-pvid
    pvid 2
    exit
!
interface port INT10
    tag-pvid
    pvid 2
    exit
!
interface port EXT10
    tagging
    pvid 2
    exit
!
vlan 2
    enable
    name "VLAN 2"
    member INT9-INT10,EXT10
!
!
vlan 300
    enable
    name "VLAN 300"
    member INT9-INT10,EXT10
!
!
vlan 400
    enable
    name "VLAN 400"
    member INT9-INT10,EXT10
```

Figure 7-8 Extract of the initial switch configuration

7.3 Initial tests

Based on the configuration as described in 7.2, “Initial configuration” on page 201, run the following tests:

- ▶ Open an internet connection from the client workstation to the Web server. Use an HTTP non-secure connection on port 80 and an HTTPS secure connection on port 443.
- ▶ Generate network traffic from the client workstation to the File server to estimate the bandwidth between the two systems.

7.3.1 Web connection tests

These tests show that the client workstation can open a connection on the Web server by using a secure or a non-secure connection. Figure 7-9 shows that HTTP connections are authorized from the client.



Figure 7-9 Home page on Web server using a non-secure connection (HTTP on port 80)

Figure 7-10 shows that HTTPS connections are authorized from the client.

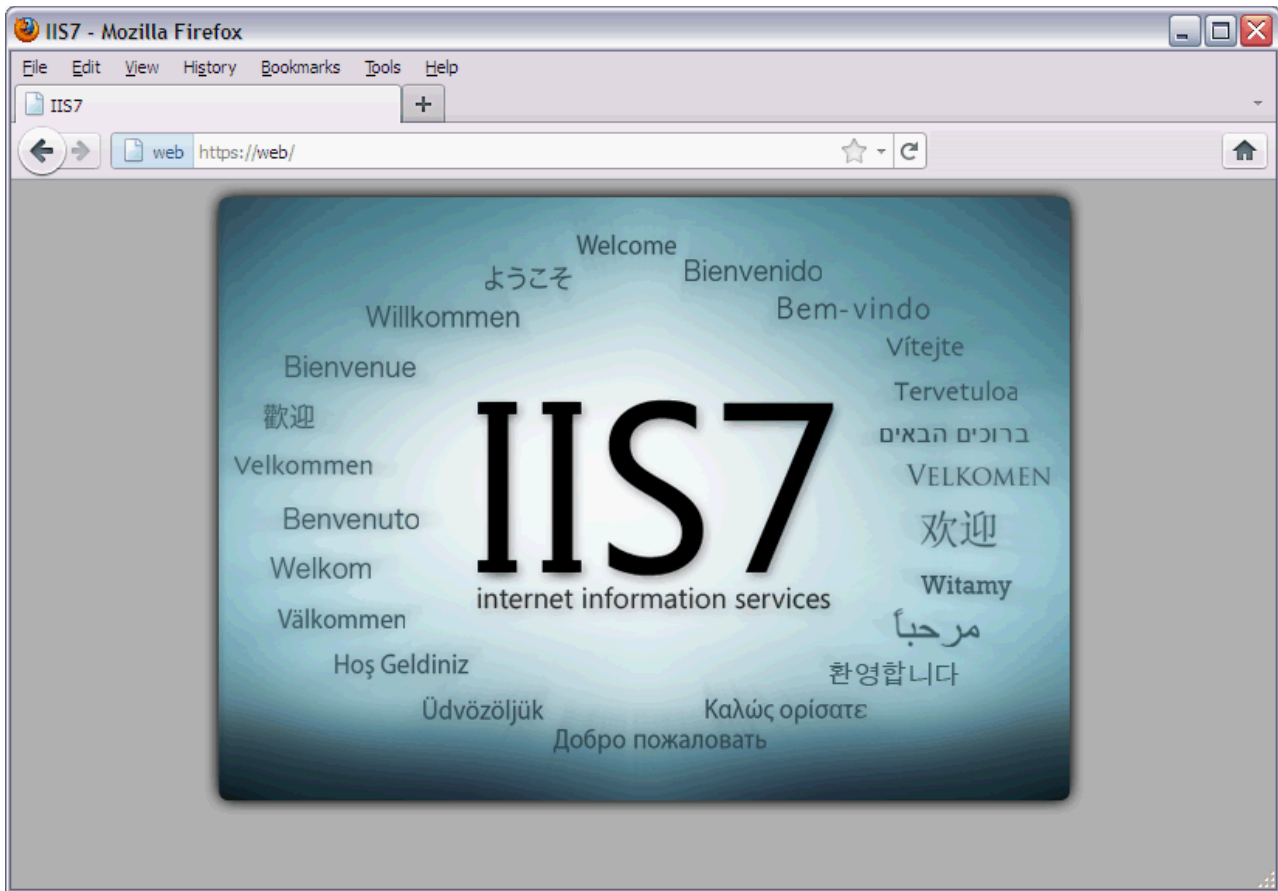


Figure 7-10 Home page on Web server using a secure connection (HTTPS on port 443)

For more information about how to create an access control list to reject the traffic that uses port 80, see 7.4.3, “Applying access control lists” on page 222.

7.3.2 Bandwidth tests

This test measures the current bandwidth between the client workstation and the File server. Set up the File server as the JPerf server and the client workstation as the JPerf client. As shown in Figure 7-11, the current bandwidth reaches 400 MBits/s.

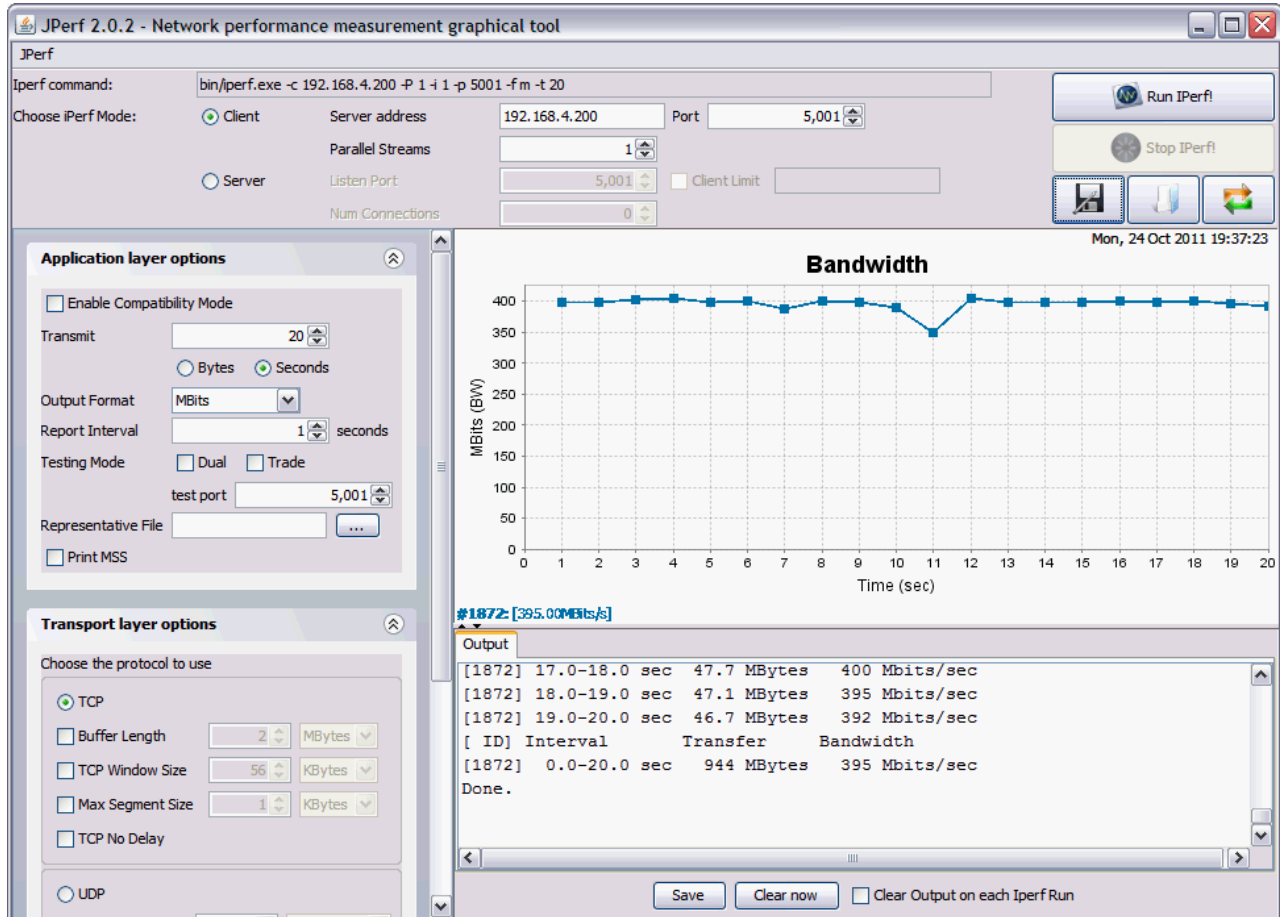


Figure 7-11 Result of JPerf test

For more information about how to configure traffic shaping to limit the bandwidth from/to a server, see 7.4.4, “Applying traffic shaping” on page 225.

7.4 Enabling VMready and implementing network policies

This section explains how VMready helps you to configure the network policies in 7.1.3, “The example scenario” on page 200. To implement the policies, you must perform these steps:

1. Enabling VMready
2. Creating the VM Groups
3. Applying access control lists
4. Applying traffic shaping

7.4.1 Enabling VMready

To be able to configure VMready on your switch and then to implement the policies, you first need to enable VMready functionality.

Enabling from the web interface

Connect to the IBM BNT Virtual Fabric 10 Gb in bay #9 with a web browser. Click the **Configuration** tab, then **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Virtualization** → **Virtual Machine** → **VM Group Port**. As shown in Figure 7-12, the switch is not yet aware of virtual machines that are hosted on the hypervisors. No virtual machines are reported on the internal ports of the switch.

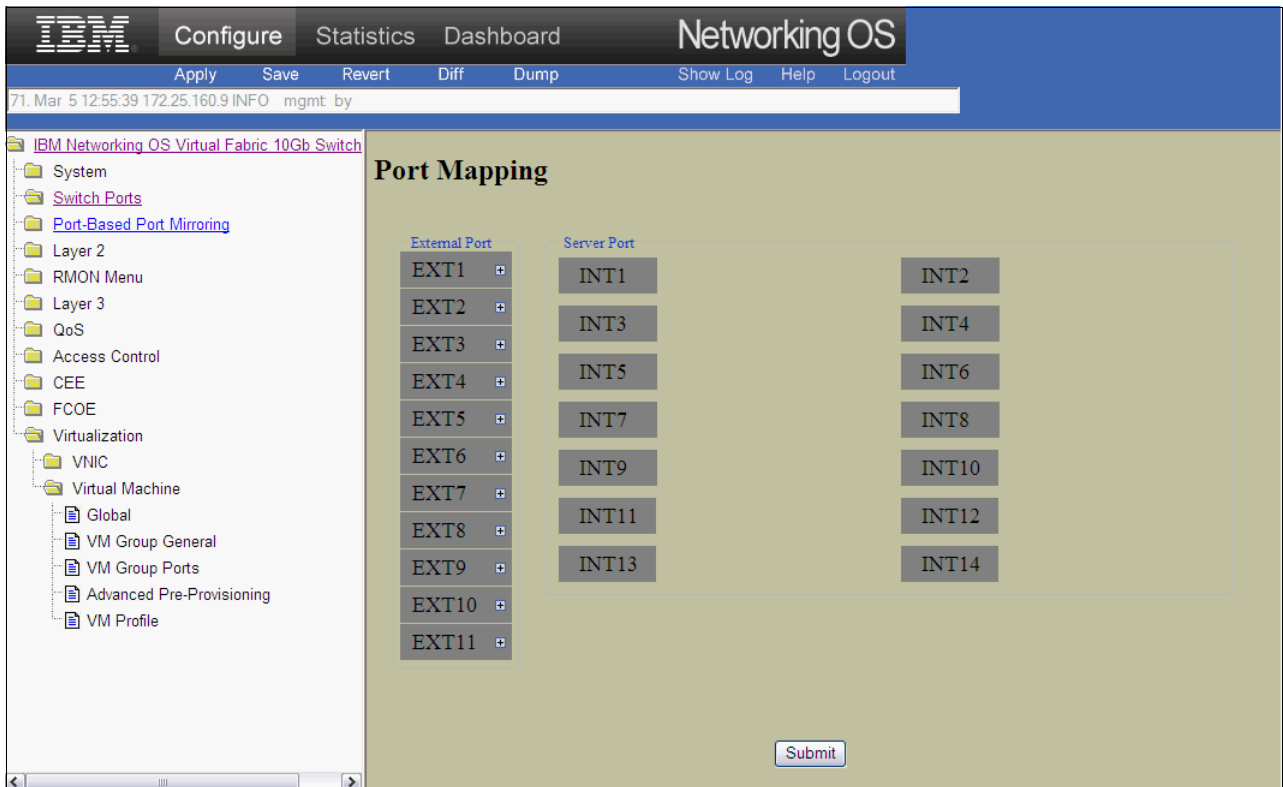


Figure 7-12 Switch internal port overview before VMready activation

To enable VMready, click **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Virtualization** → **Virtual Machine** → **Global**. In the **Virtual Machine Groups** menu, select **Enabled** as shown in Figure 7-13. Then click **Submit** → **Apply** → **Save**.

Attention: For all configuration steps, you must click **Submit** to save a change, **Apply** to activate it, and then **Save** to keep your configuration active after a switch reboot.

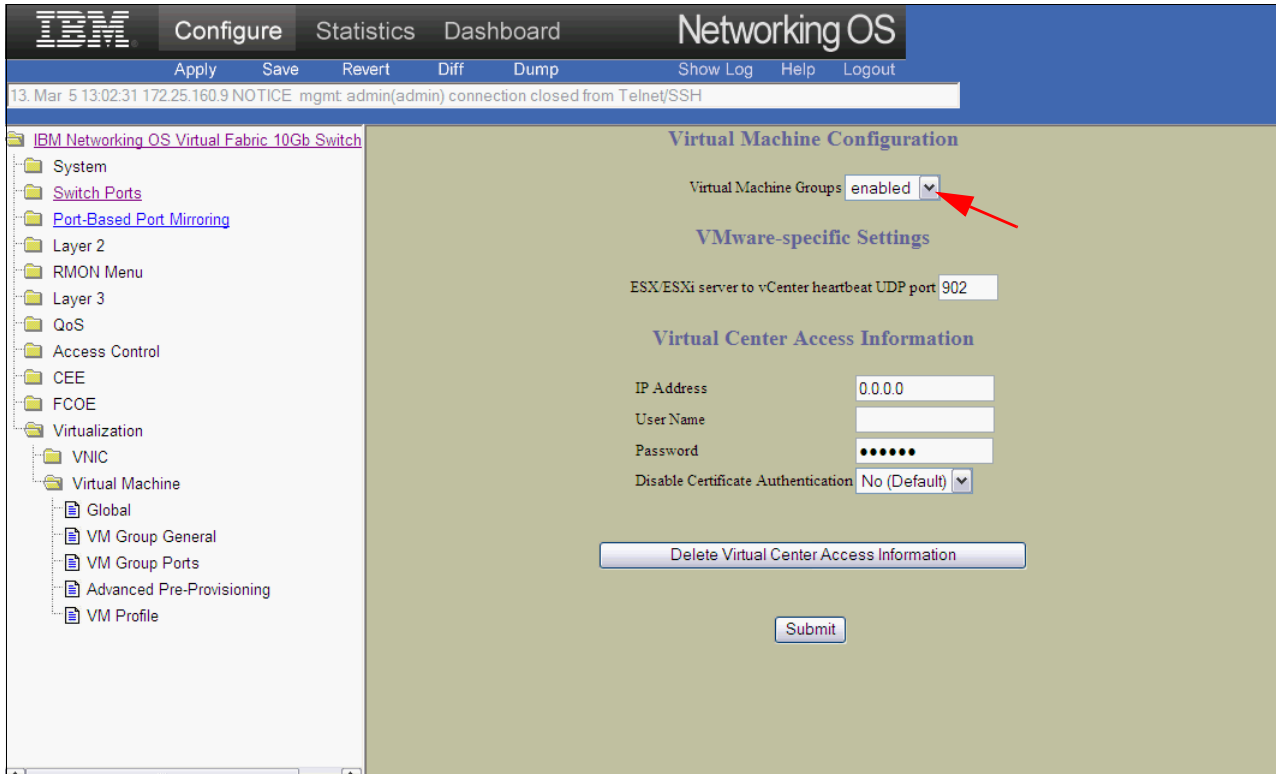


Figure 7-13 Enabling VMready

Click **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Virtual Machine** → **VM Group Ports**. You can now see that the switch has discovered virtual machines running on the hypervisors. Figure 7-14 shows that on port INT9 and port INT10 of the switch, it has discovered the three virtual machines. You can identify them by their IP addresses.

Tip: To discover the virtual machines, you need to generate ingoing or outgoing network traffic for each of the virtual machines. Internal host network traffic (from one VM to another VM on the same hypervisor) does not generate traffic on the physical switch.

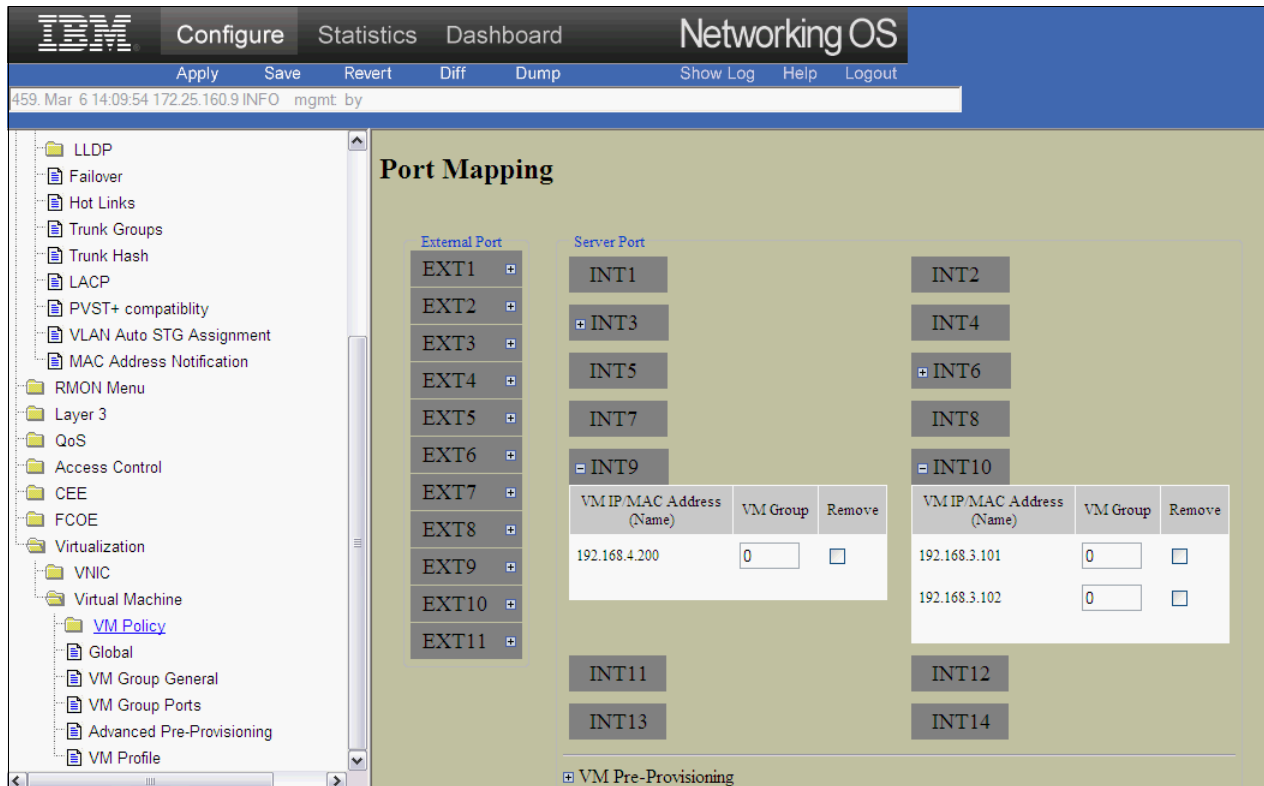


Figure 7-14 VM Group Ports overview after VMready is enabled

Enabling from the CLI

To enable VMready, connect to the IBM BNT Virtual Fabric 10 Gb Switch Module in bay #9 of the chassis. Figure 7-15 shows the `virt enable` command enabling VMready from the CLI.

```
Router>enable
Enable privilege granted.

Router#conf terminal
Enter configuration commands, one per line. End with Ctrl/Z.

Router(config)#virt enable
Router(config)#^Z

Router#write
```

Figure 7-15 Enabling VMready from the CLI

7.4.2 Creating the VM Groups

The scenario uses a production environment where VLAN #300 and VLAN #400 already exist. To use VMready functions such as implementing access control list, associate the virtual machines to VM Groups.

VM Groups are containers that substitute to the VLAN entities. A VM Group encapsulates the VLAN entity. VM Groups and VLANs use the same table of IDs, so they cannot coexist. To implement one VM Group with a VLAN ID, that VLAN ID must be removed from the VLAN table.

In this example, you need to remove VLAN #300 and VLAN #400 from the switch configuration. Then you need to create two new VM Groups #3 and #4 that are associated to VLAN #300 and VLAN #400. Finally, associate the virtual machines to these VM Groups. The Web and DB servers were set up in VLAN #300, so it becomes a member of VM Group #3. The File server was set up in VLAN #400, so it becomes a member of VM Group #4.

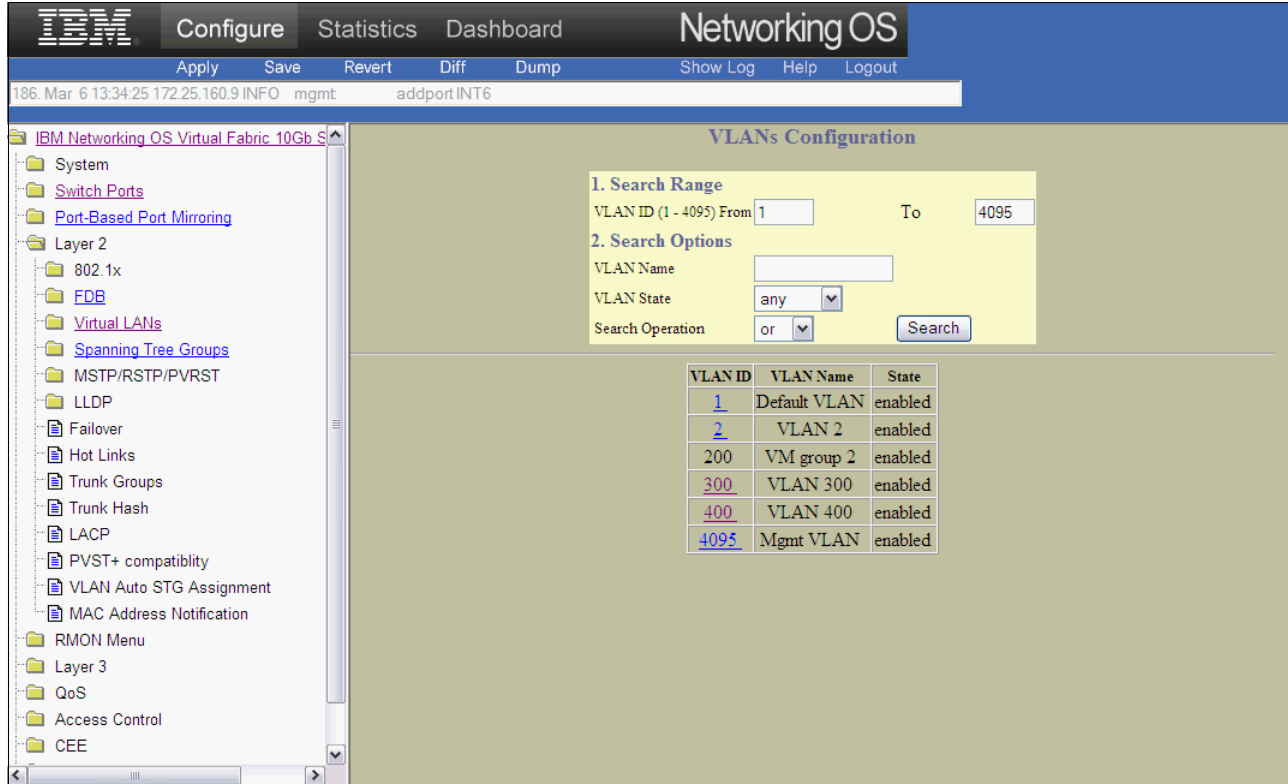
Consideration: VLANs that are used with virtual machines are not defined in the usual VLAN table, but in the VM Group table with VMready enabled. The IDs of the VLANs and the IDs of the VM Groups are common. You cannot use the same ID to create a VLAN and to create a VM Group.

By default when you create a VM Group, a new VLAN ID is automatically created in the VLAN table. This VLAN has the same ID as the VM Group and is a reference to this new VM Group. You cannot modify the VLAN properties of a VM Group from the VLAN table. You must use the VM Group properties.

Attention: If you are in a production environment, make sure that you have no activity on servers that are members of the VLAN that you are about to remove. Otherwise the servers will lose their connectivity.

Creating VM Groups from the web interface

Before creating the VM Groups, you must remove the current VLAN configuration. Open a web browser to the IBM BNT Virtual Fabric 10 Gb switch module. Click the Configuration tab, then **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Layer 2** → **Virtual LANs**, as shown in Figure 7-16. Select VLAN #300 and delete it. Then do the same for VLAN #400.



The screenshot displays the 'VLANs Configuration' page in the IBM Networking OS Virtual Fabric 10Gb Switch Module web interface. The interface includes a navigation menu on the left, a top navigation bar with 'Configure', 'Statistics', and 'Dashboard' tabs, and a main content area. The main content area shows search options for VLANs, including a search range (1-4095), a search operation dropdown (set to 'or'), and a search button. Below the search options is a table listing existing VLANs.

VLAN ID	VLAN Name	State
1	Default VLAN	enabled
2	VLAN 2	enabled
200	VM group 2	enabled
300	VLAN 300	enabled
400	VLAN 400	enabled
4095	Mgmt VLAN	enabled

Figure 7-16 Virtual LANs configuration

The next step is to create the VM Groups. Click **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Virtualization** → **Virtual Machines** → **VM Group General**, as shown in Figure 7-17.

Remember: Physical servers running bare metal operating systems without virtualization that are members of the VLAN configuration you have removed require special attention. You need to assign them back into the VM Group. To achieve that, use the **Port add** field in the VM Group configuration page and add your internal or external ports.

Do not add the ports of your hypervisor that support the virtual machines in the VM Groups. VMready automatically configures the VLANs on the ports that support the virtual machines network.

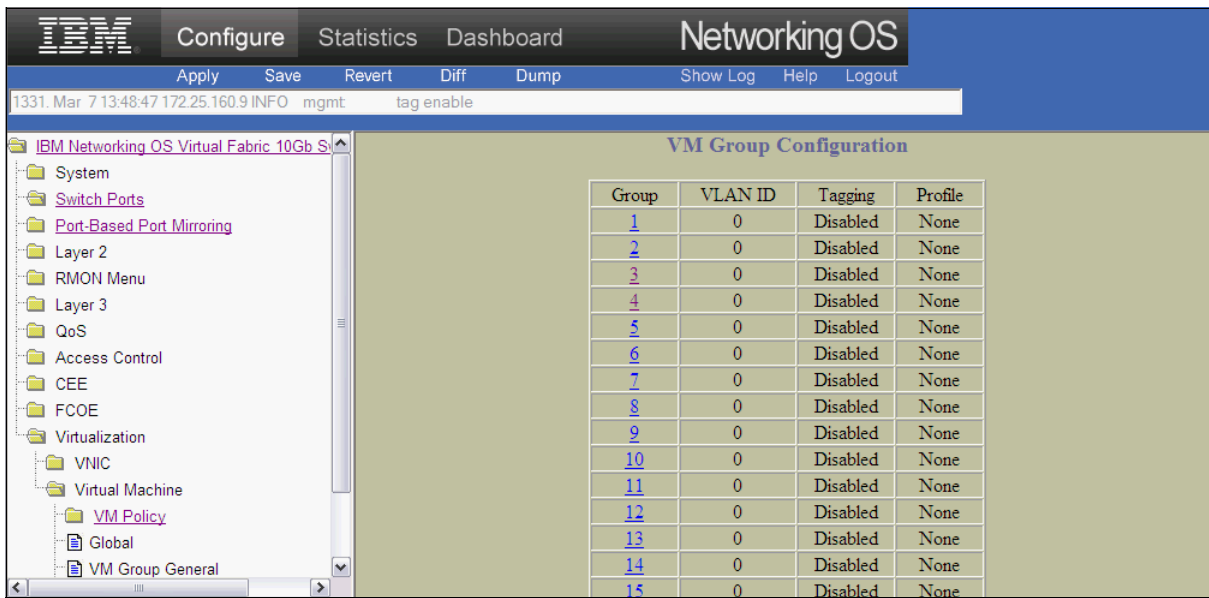


Figure 7-17 VM Groups overview

Click VM Group #3, which is not yet defined. Assign VLAN #300 to this entity and enable tagging as shown in Figure 7-18. Repeat the same action for VLAN #400.

Consideration: At this point of the setup, you cannot reach the virtual machines from the client. The physical ports on the switch are not aware of VLAN #300 and VLAN #400 yet. Therefore, all the traffic from or to the virtual machines is blocked.

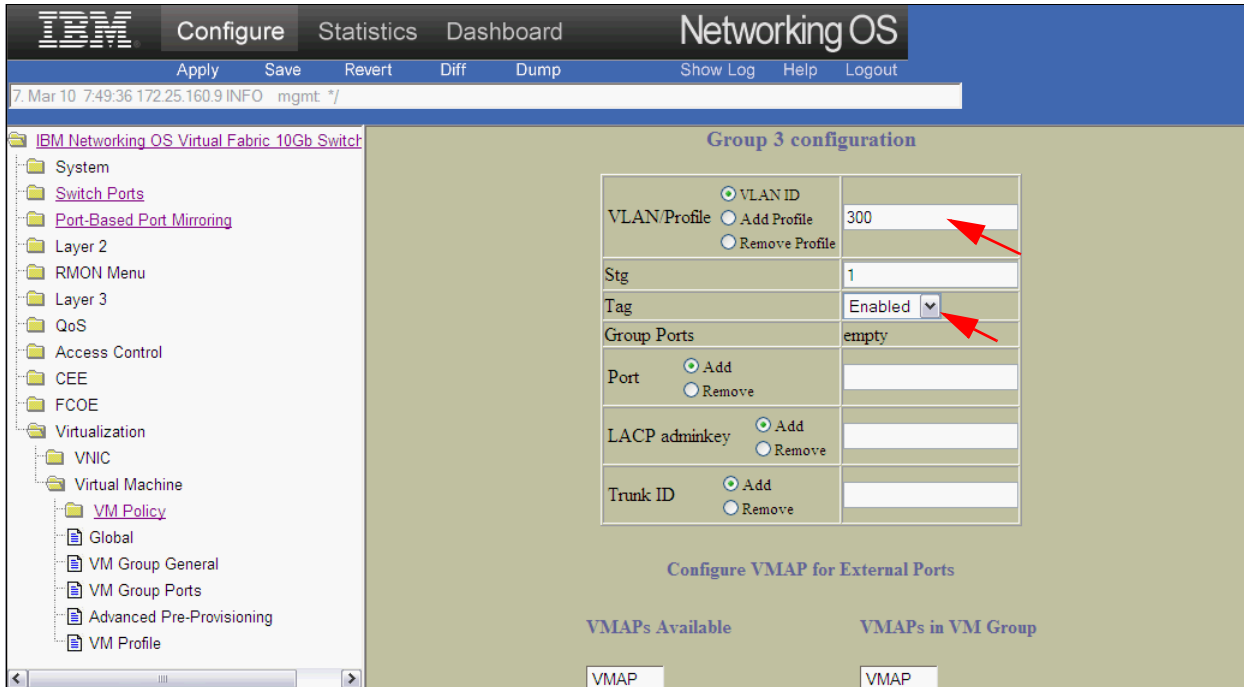


Figure 7-18 Creating VM groups for VLAN #300

After the creation of the VM Groups, the virtual machines need to be added to these groups. You need the MAC addresses of your virtual machines to associate them to the groups you created. Click the **Configuration** tab, then **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Virtualization** → **Virtual Machines** → **VM Group General**. As shown in Figure 7-19, enter each MAC address and associate them to their related VM Groups:

- ▶ File server is associated with VM Group #4 (VLAN #400)
- ▶ Web server is associated with VM Group #3 (VLAN #300)
- ▶ DB server is associated with VM Group #3 (VLAN #300)

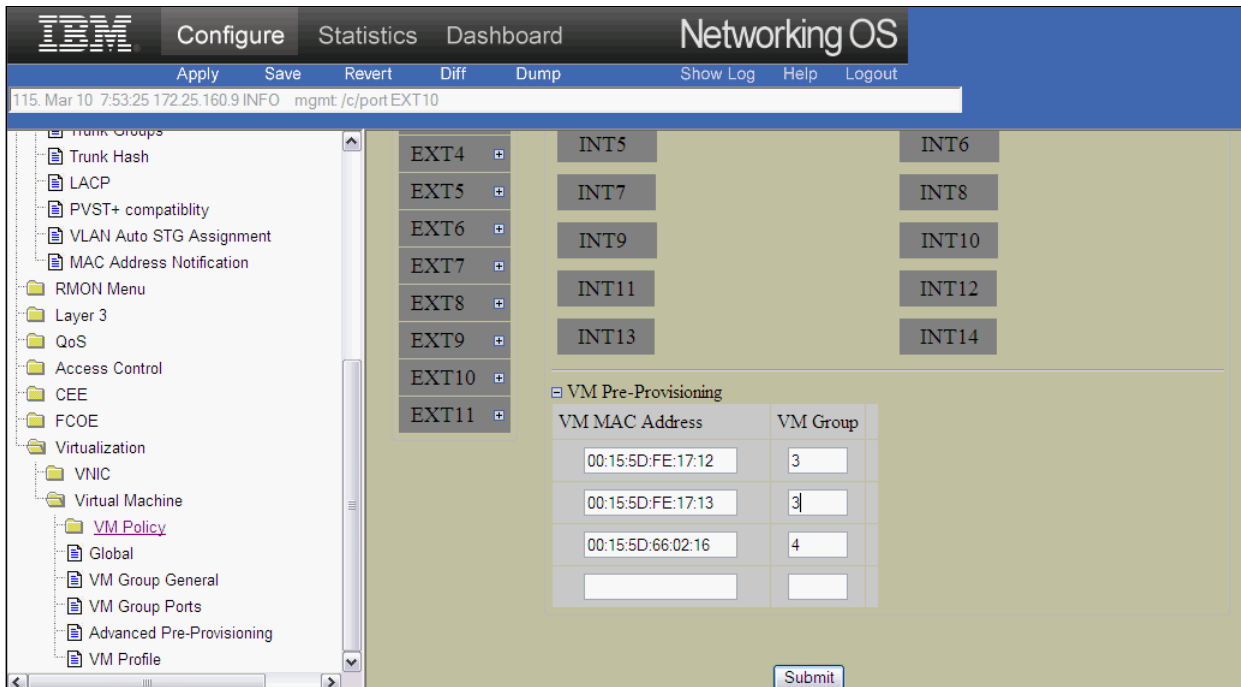


Figure 7-19 Pre-provisioning of the MAC addresses

After this step, you can generate traffic between your client workstation and the virtual machines. Figure 7-20 shows that port INT9 and INT10 are members of VLAN #2, which is the private VLAN ID. INT9 and port INT10 were assigned some additional VLANs, which are VLAN #300 and VLAN #400. These VLANs were automatically associated with these ports because the switch discovered some network traffic coming from the virtual machines.

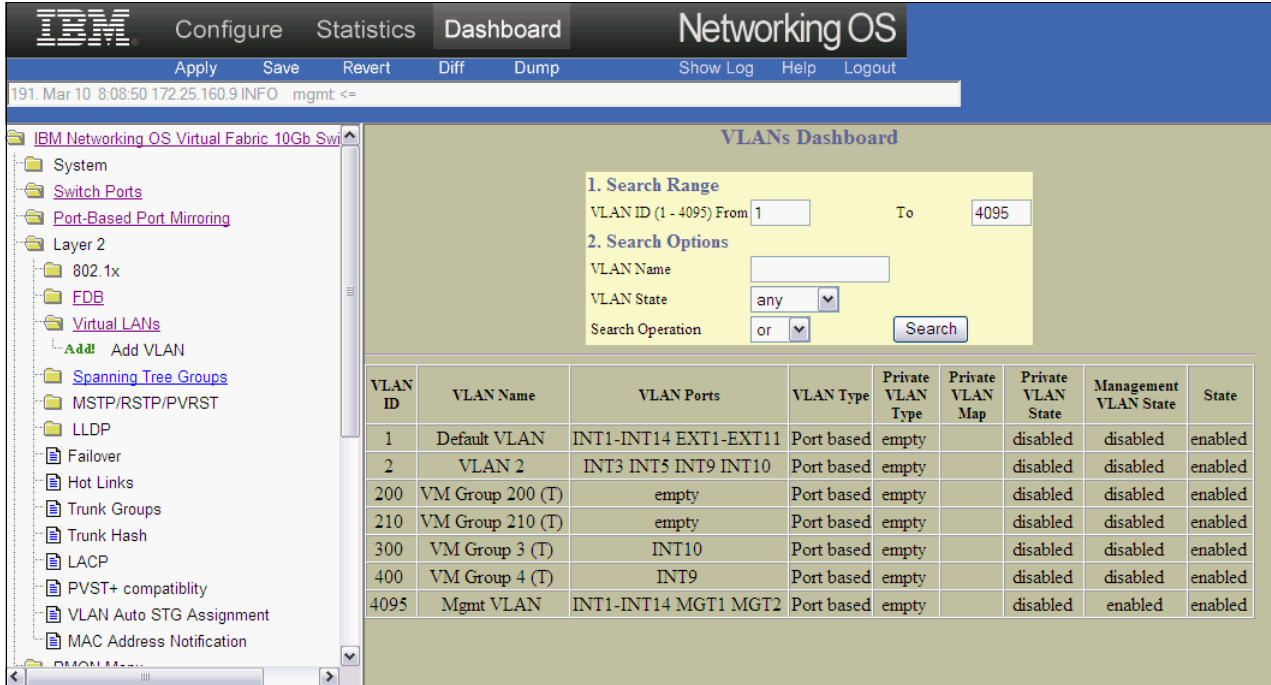


Figure 7-20 Port configuration after some network traffic

Creating VM Groups from the CLI

This section addresses these steps:

1. Removing the existing VLANs
2. Creating the VM Groups and associate the virtual machines to these groups

Figure 7-21 shows how to remove the current VLANs #300 and #400 from the configuration.

```
Router>show vlan
VLAN          Name                Status MGT          Ports
-----
1             Default VLAN       ena   dis   INT1-INT14 EXT1-EXT11
2             VLAN 2             ena   dis   INT9 INT10 EXT10
300 VLAN 300   ena   dis   INT9 INT10 EXT10
400 VLAN 400   ena   dis   INT9 INT10 EXT10
4095 Mgmt VLAN     ena   ena   INT1-INT14 MGT1 MGT2
Router>enable

Enable privilege granted.
Router#conf t
Enter configuration commands, one per line.  End with Ctrl/Z.
Router(config)#no vlan 300
Router(config)#no vlan 400
Router(config)#^Z

Router#write
```

Figure 7-21 Removing the current VLANs

Figure 7-22 shows how to create VM Groups #3 for VLAN #300 and VM Group #4 for VLAN #400, and then associate them.

```
Router>enable
Enable privilege granted.

Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.

Router(config)#virt vmgroup 3 vlan 300
VLAN 300 was assigned to STG 44.
Router(config)#virt vmgroup 3 tag

Router(config)#virt vmgroup 4 vlan 400
VLAN 400 was assigned to STG 16.
Router(config)#virt vmgroup 4 tag

Router(config)#virt vmgroup 3 vm 00:15:5D:FE:17:12
INFO: 00:15:5d:fe:17:12 is not in switch FDB.

Router(config)#virt vmgroup 3 vm 00:15:5D:FE:17:13
INFO: 00:15:5d:fe:17:13 is not in switch FDB.

Router(config)#virt vmgroup 4 vm 00:15:5D:66:02:16
INFO: 00:15:5d:66:02:16 is not in switch FDB.

Router(config)#^Z

Router#write
```

Figure 7-22 Creating the VM Groups and associating the virtual machines

After the VMs are assigned to a VM Group, generate traffic so you can see them in the configuration. Figure 7-23 shows the discovery of the three virtual machines.

```

Mar  7 14:05:06 172.25.160.9 NOTICE  vm: Virtual Machine with IP address
192.168.4.200 came online

Router#show virt vm
IP Address          VMAC Address          Index Port    VM Group (Profile)
-----
192.168.4.200      00:15:5d:66:02:16    6   INT9     4
0.0.0.0            00:15:5d:fe:17:13    3   UNKNOWN 3
0.0.0.0            00:15:5d:fe:17:12    2   UNKNOWN 3

Number of entries: 3
0.0.0.0 indicates IP address not yet available

Mar  7 14:08:12 172.25.160.9 NOTICE  vm: Virtual Machine with IP address
192.168.3.101 came online
Mar  7 14:08:15 172.25.160.9 NOTICE  vm: Virtual Machine with IP address
192.168.3.102 came online

Router#show virt vm
IP Address          VMAC Address          Index Port    VM Group (Profile)
-----
192.168.4.200      00:15:5d:66:02:16    6   INT9     4
192.168.3.102      00:15:5d:fe:17:12    2   INT10    3
192.168.3.101      00:15:5d:fe:17:13    8   INT10    3

Number of entries: 3

```

Figure 7-23 Virtual machines discovery

7.4.3 Applying access control lists

In the scenario, you want to block all the traffic of the non-secure connections that go to the Web server. To do so, create a VLAN Map (VMAP) to block all packets that go to the Web server on port 80. A VMAP is like an access control list (ACL), but it is applied to a VM Group. The VMAP is associated to the VM Group to which the VM related to the VMAP belongs.

This section addresses the following steps:

1. Creating a VMAP that blocks the traffic for the Web server on port 80
2. Associating this VMAP to VM Group #3

Applying ACL from the web interface

Connect to the IBM BNT Virtual Fabric 10 Gb in bay #9 with a web browser. Click the **Configuration** tab, then **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Access Control** → **VLAN MAP** → **Add VMAP**. As shown in Figure 7-24, you can specify multiples restrictions on a particular VM or groups of VMs. In this example, you want to block all the traffic that goes to the Web server on port 80. Set **Action** to **Deny**, enter the **Destination IP Address** of the Web server, and select **Destination Port 80**.

Field	Value	Mask
VMAP Id (1 - 128)	3	
Group Id	0	
Filter Action	Deny	
Set priority value	none	
Ethernet Packet Format	Disabled	
Tagging Packet Format	Disabled	
IP Packet Format	None	
Source MAC Address	00:00:00:00:00:00	ff:ff:ff:ff:ff:ff
Destination MAC Address	00:00:00:00:00:00	ff:ff:ff:ff:ff:ff
Ethernet Type	None	Value (0600-fff) 600
VLAN Id (1-4095, 0 = any)	1	Mask (0-fff) fff
802.1p Priority	None	
Type of Service (0-255)	0	Disabled
Protocol (0-255)	0	Disabled
Source IP Address	0.0.0.0	255.255.255.255
Destination IP Address	192.168.3.101	255.255.255.255
TCP/UDP Src Port (1-65535)	1	Mask (1-fff) fff
TCP/UDP Dst Port (1-65535)	80	Mask (1-fff) fff
TCP Flags	<input type="checkbox"/> FIN <input type="checkbox"/> SYN <input type="checkbox"/> RST <input type="checkbox"/> PSH <input type="checkbox"/> ACK <input type="checkbox"/> URG	Mask(0-3f) 3f
Statistics	Disabled	
Mirror Port	INT1	
Egress port	None	

Figure 7-24 VMAP configuration

After creating the VMAP, associate it to the VM Group that contains the Web virtual machine. To do so, click **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Virtualization** → **Virtual Machine** → **VM Group General**. As shown in Figure 7-25, enter VM Group #3 to edit the configuration. Add your VMAP in the **Configure VMAPs for All Ports** list.

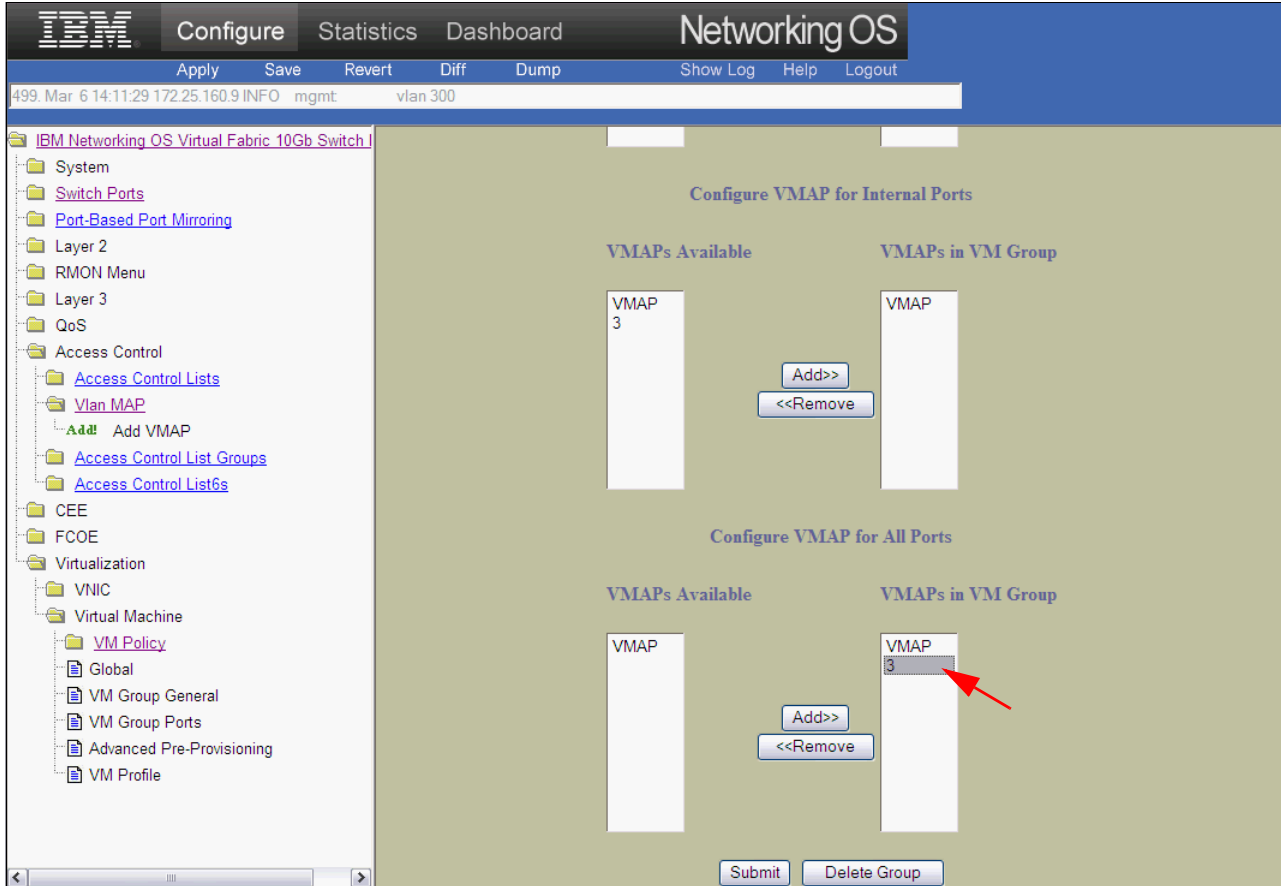


Figure 7-25 VMAP association within the VM Group

Applying ACL from the CLI

Connect to the IBM BNT Virtual Fabric Switch Module in bay #9 using a telnet session. As shown in Figure 7-26, **access-control vmap** commands creates the VMAP and **virt vmgroup** command associate the VMAP to a VM Group.

```
Router>enable
Enable privilege granted.

Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.
Router(config)#

Router(config)#access-control vmap 3 action deny

Router(config)#access-control vmap 3 ipv4 destination-ip-address 192.168.3.101

Router(config)#access-control vmap 3 tcp-udp destination-port 80

Router(config)#virt vmgroup 3 vmap 3
Router(config)#^Z

Router#write
```

Figure 7-26 Applying VMAP with the CLI

7.4.4 Applying traffic shaping

In this scenario, you want to restrict the bandwidth which is allocated to the File server. To do that, create a VM Policy and apply a traffic shaping on the incoming and outgoing traffic of the virtual machine.

Remember: The policy applies only to traffic that goes through the physical switch. It does not apply between two virtual machines on the same host whose traffic stays in the virtual switch.

Applying traffic shaping from the web interface

Connect to the IBM BNT Virtual Fabric 10 Gb in bay #9 with a web browser. Click the **Configuration** tab, then **IBM Networking OS Virtual Fabric 10Gb Switch Module** → **Virtualization** → **Virtual Machine** → **VM Policy** → **Add VM Policy**. As shown in Figure 7-27, enter the MAC address of the File server and activate the Bandwidth Control. Set a maximum rate of 1 MB/s for both incoming and outgoing traffic. Finally, select an ACL ID that is not used, number #4 in this example. After applying, the policy is immediately taken into account.

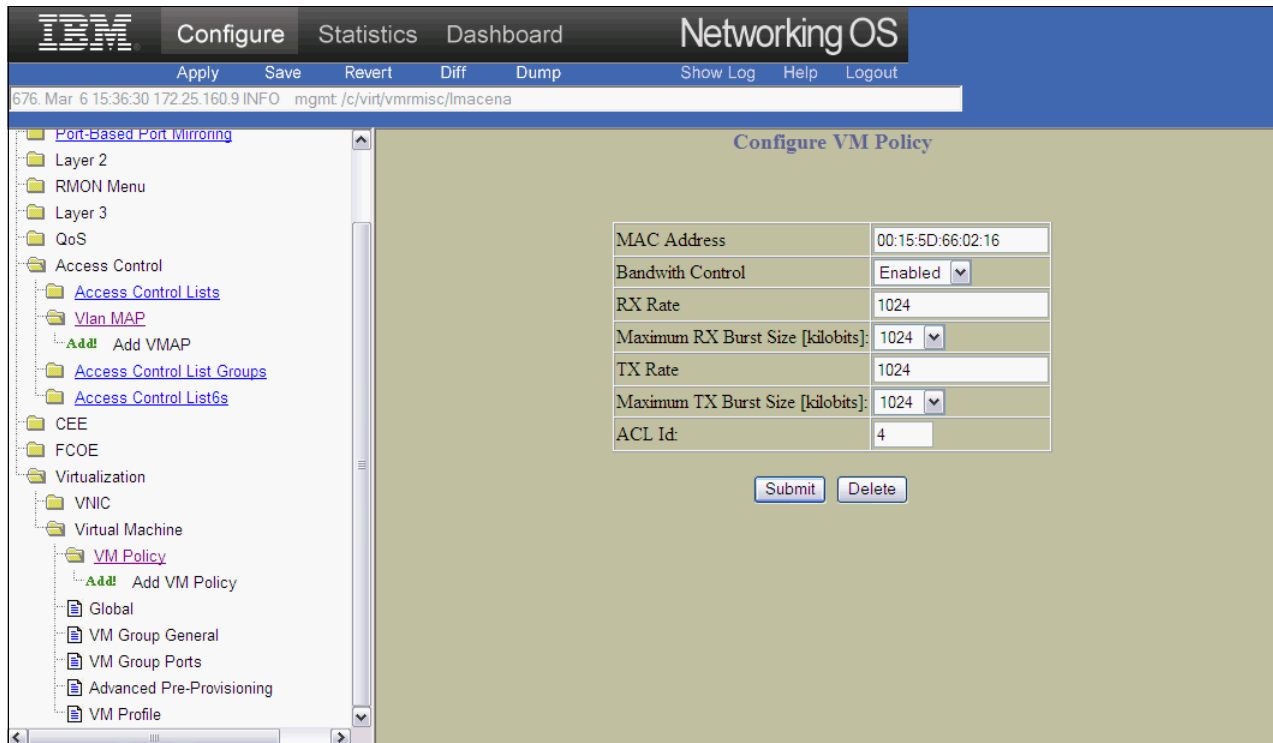


Figure 7-27 Traffic shaping configuration

Applying traffic shaping from the CLI

Figure 7-28 shows how the `virt vmpolicy vmbwidth` command creates a VM Policy for traffic shaping.

```
Router>enable
Enable privilege granted.
Router#
Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.

Router(config)#virt vmpolicy vmbwidth 00:15:5D:66:02:16 txrate 1024 1024 4
Router(config)#virt vmpolicy vmbwidth 00:15:5D:66:02:16 rxrate 1024 1024
Router(config)#virt vmpolicy vmbwidth 00:15:5D:66:02:16 bwctrl
Router(config)#^Z

Router#write
```

Figure 7-28 Applying traffic shaping

7.5 Validating the network policies

Now that all the network policies are enabled, validate that they are applied. This section addresses the following steps:

1. Validate the ACL policy
2. Validate the traffic shaping policy

7.5.1 Validating the ACL policy

To check that your VMAP is correctly configured, try to open a non-secure connection on your Web server. Also try to open a secure connection to make sure that the traffic is not blocked by the switch. As shown in Figure 7-29, the traffic is blocked on port 80 of the Web server.

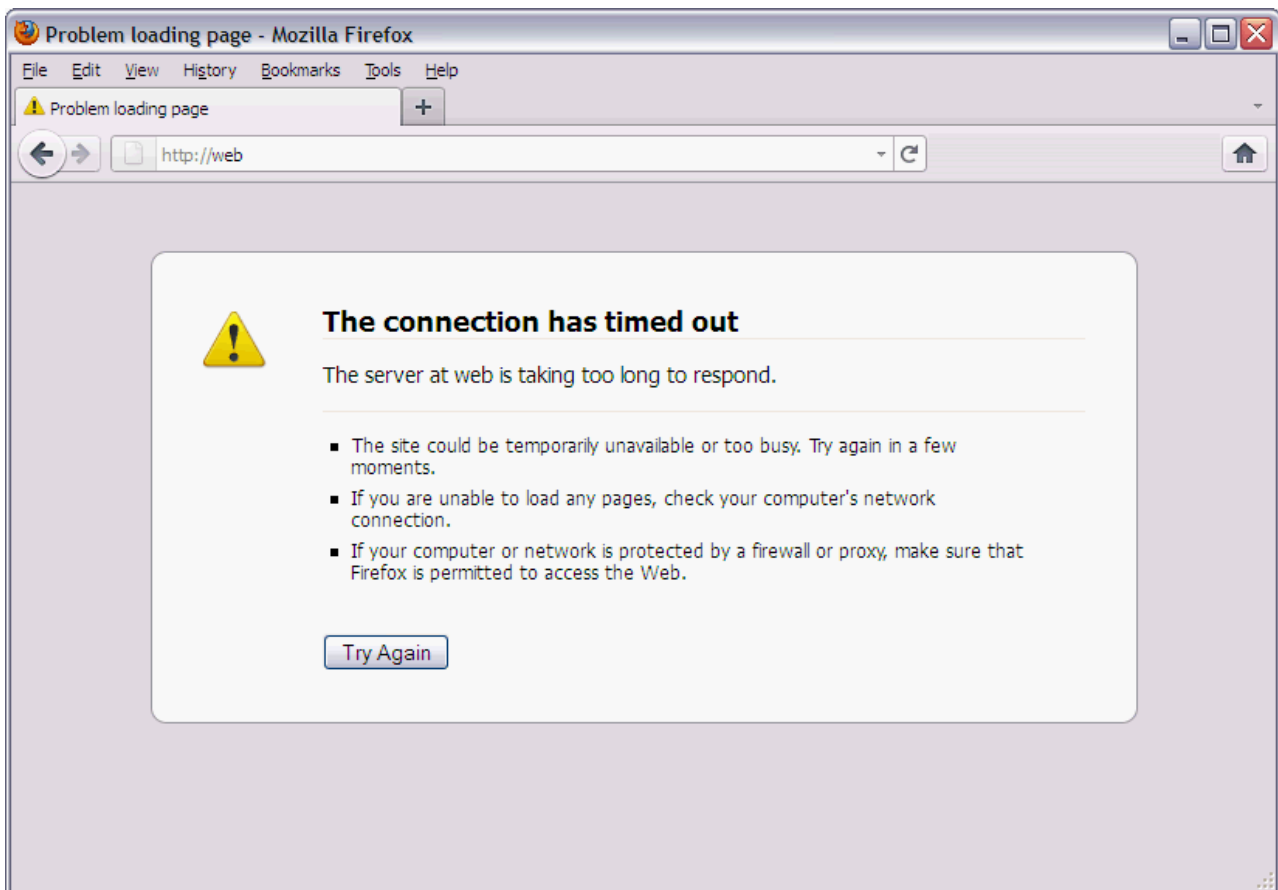


Figure 7-29 Home page on Web server using a non-secure connection

Figure 7-30 shows that Web server still responds to ping but network traffic on port 80 is automatically blocked by the switch.

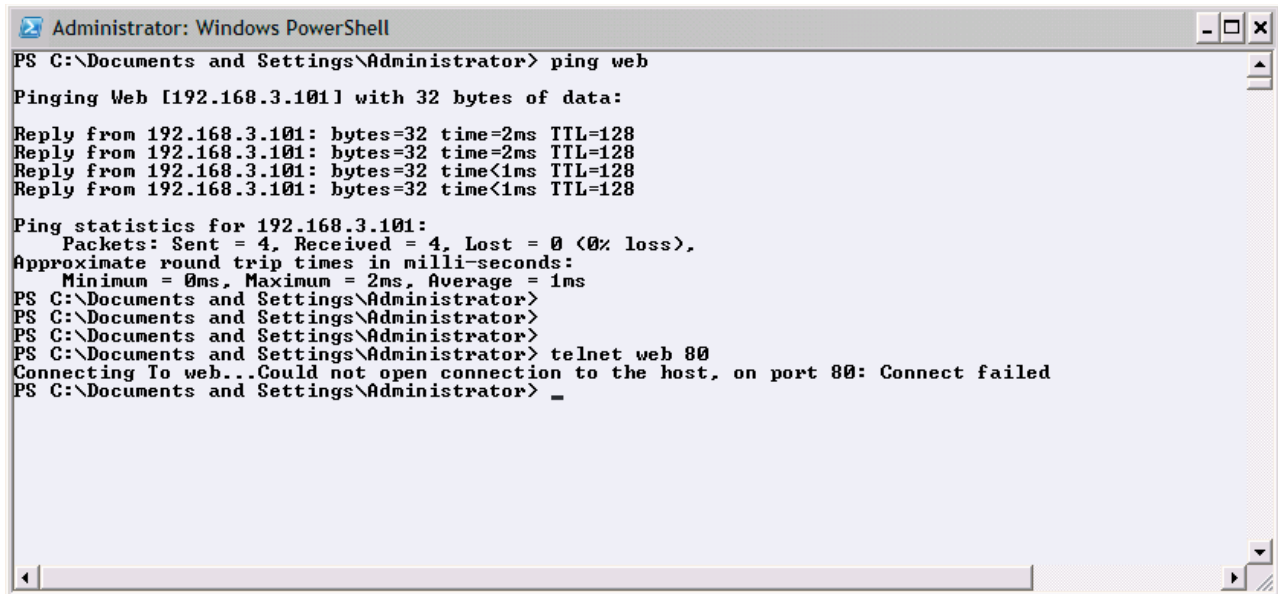


Figure 7-30 Telnet connection on Web server on port 80

Figure 7-31 confirms that secure traffic is still active.



Figure 7-31 Home page on Web server using a secure connection

7.5.2 Validating the traffic shaping policy

To validate your network policy based on traffic shaping between your client workstation and File server, run a JPerf test. Figure 7-32 shows that the network is now limited to 1 MBits/s.

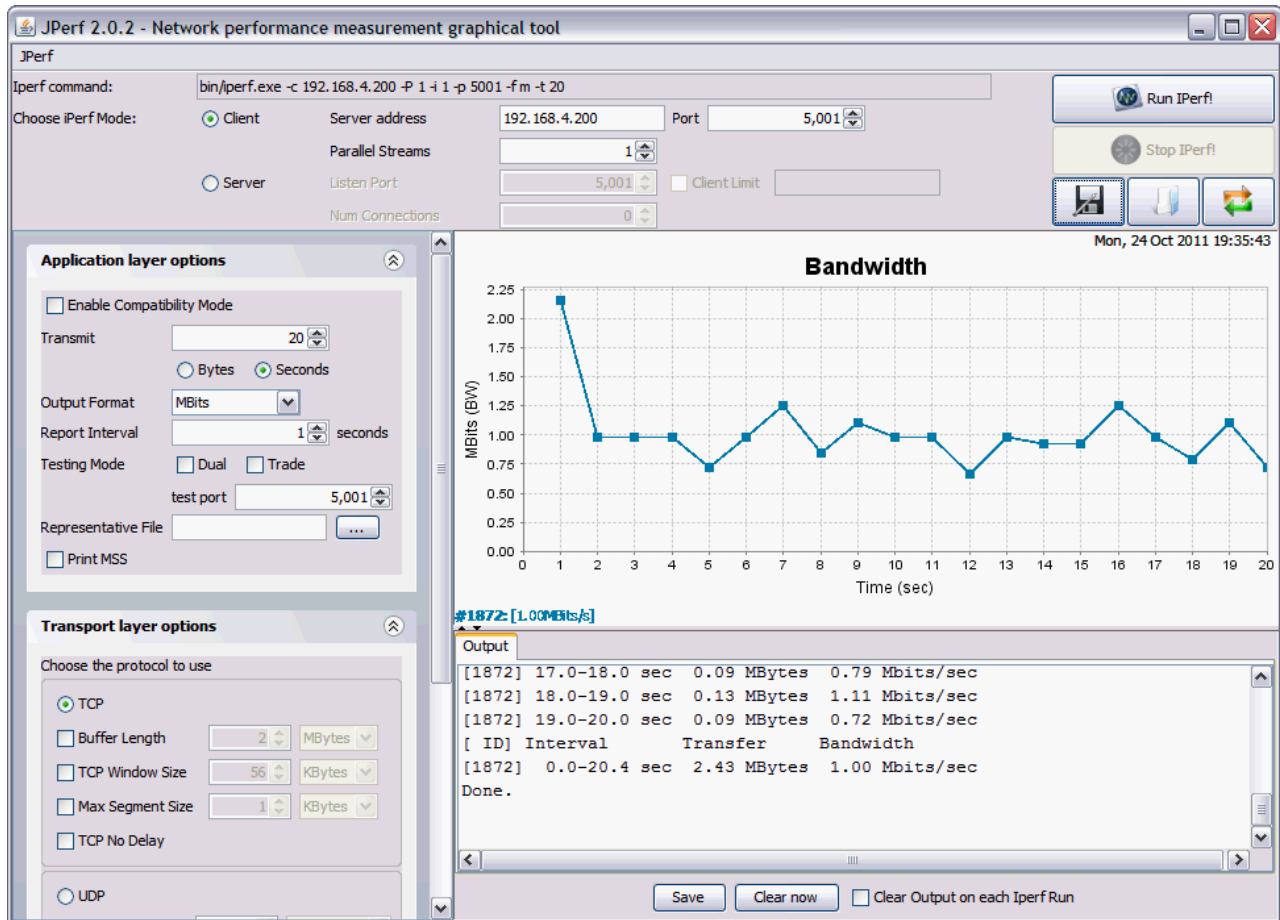


Figure 7-32 JPerf test after network policy

7.6 Validating Nmotion

After showing that the network policies have been applied, validate that they are still active after a VM migration from one host to another.

During the initial tests, the File server was hosted on HyperV1 (INT9 on the switch), and the Web and DB servers were hosted on HyperV2 (INT10 on the switch). As shown in Figure 7-33, port INT9 has been automatically configured to allow VLAN #400. INT10 has been automatically configured to allow VLAN #300.

```

Router>sho vlan
VLAN          Name                Status MGT          Ports
-----
1      Default VLAN      ena   dis   INT1-INT14 EXT1-EXT11
2      VLAN 2            ena   dis   INT9 INT10 EXT10
300  VM Group 3 (T)    ena   dis   INT10 EXT10
400  VM Group 4 (T)    ena   dis   INT9  EXT10
4095  Mgmt VLAN         ena   ena   INT1-INT14 MGT1 MGT2
Router>
Router>sho virt vm
IP Address      VMAC Address      Index Port    VM Group (Profile)
-----
192.168.4.200   00:15:5d:66:02:16 6    INT9    4
192.168.3.102   00:15:5d:fe:17:12 2    INT10   3
192.168.3.101   00:15:5d:fe:17:13 8    INT10   3

Number of entries: 3
Router>

```

Figure 7-33 VLAN configuration overview before live migration

Migrate File virtual machine from HyperV1 to HyperV2 and Web virtual machine from HyperV2 to HyperV1, as shown in Figure 7-34.

Name	Status	Type	Current Ow...	Auto start
DB	Online	Virtual Machi...	HyperV2	Yes
File	Online	Virtual Machi...	HyperV2	Yes
Web	Online	Virtual Machi...	HyperV1	Yes

Figure 7-34 Failover cluster overview

Figure 7-35 shows that the port INT9 and INT10 have been reconfigured so they can allow VLAN #300 and VLAN #400 traffic.

```

Router>sho vlan
VLAN          Name                Status MGT          Ports
-----
1             Default VLAN        ena   dis   INT1-INT14 EXT1-EXT11
2             VLAN 2              ena   dis   INT9 INT10 EXT10
300 VM Group 3 (T)    ena   dis   INT9 INT10 EXT10
400 VM Group 4 (T)    ena   dis   INT9 INT10 EXT10
4095         Mgmt VLAN          ena   ena   INT1-INT14 MGT1 MGT2
Router>
Router>sho virt vm
IP Address    VMAC Address        Index Port    VM Group (Profile)
-----
192.168.4.200 00:15:5d:66:02:16  6   INT10    4
192.168.3.102 00:15:5d:fe:17:12  2   INT10    3
192.168.3.101 00:15:5d:fe:17:13  8   INT9     3

Number of entries: 3
Router>

```

Figure 7-35 VLAN configuration overview after live migration

7.6.1 Validating the ACL policy

Run the tests again to prove that the network policy has been moved with Web server migration. Figure 7-36 shows that port 80 is still blocked.

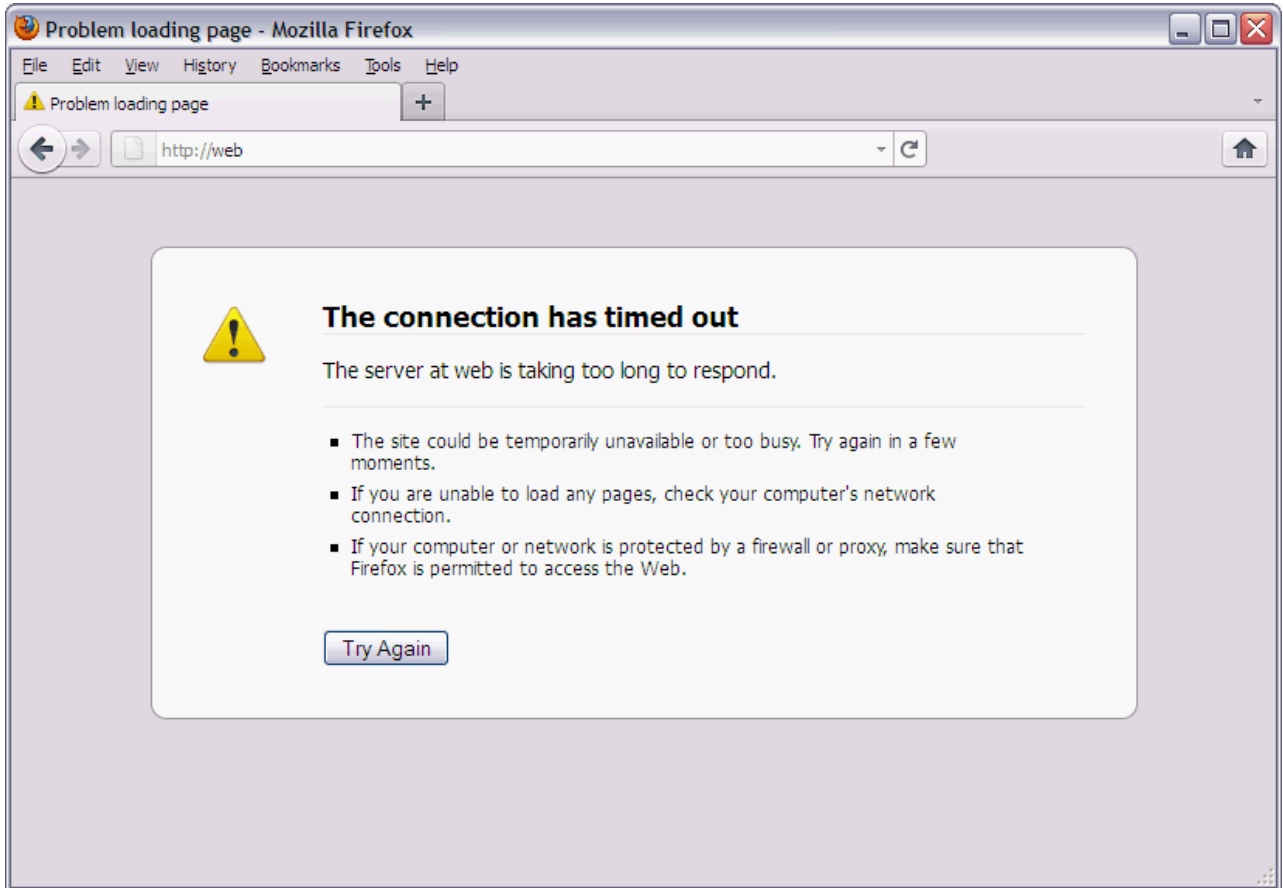


Figure 7-36 Non-secure connection on Web server

7.6.2 Validating the traffic shaping policy

Figure 7-37 shows the network traffic on File server before, during and after the migration. At 27s, the virtual machine migration is complete. The policy is still active after the migration.

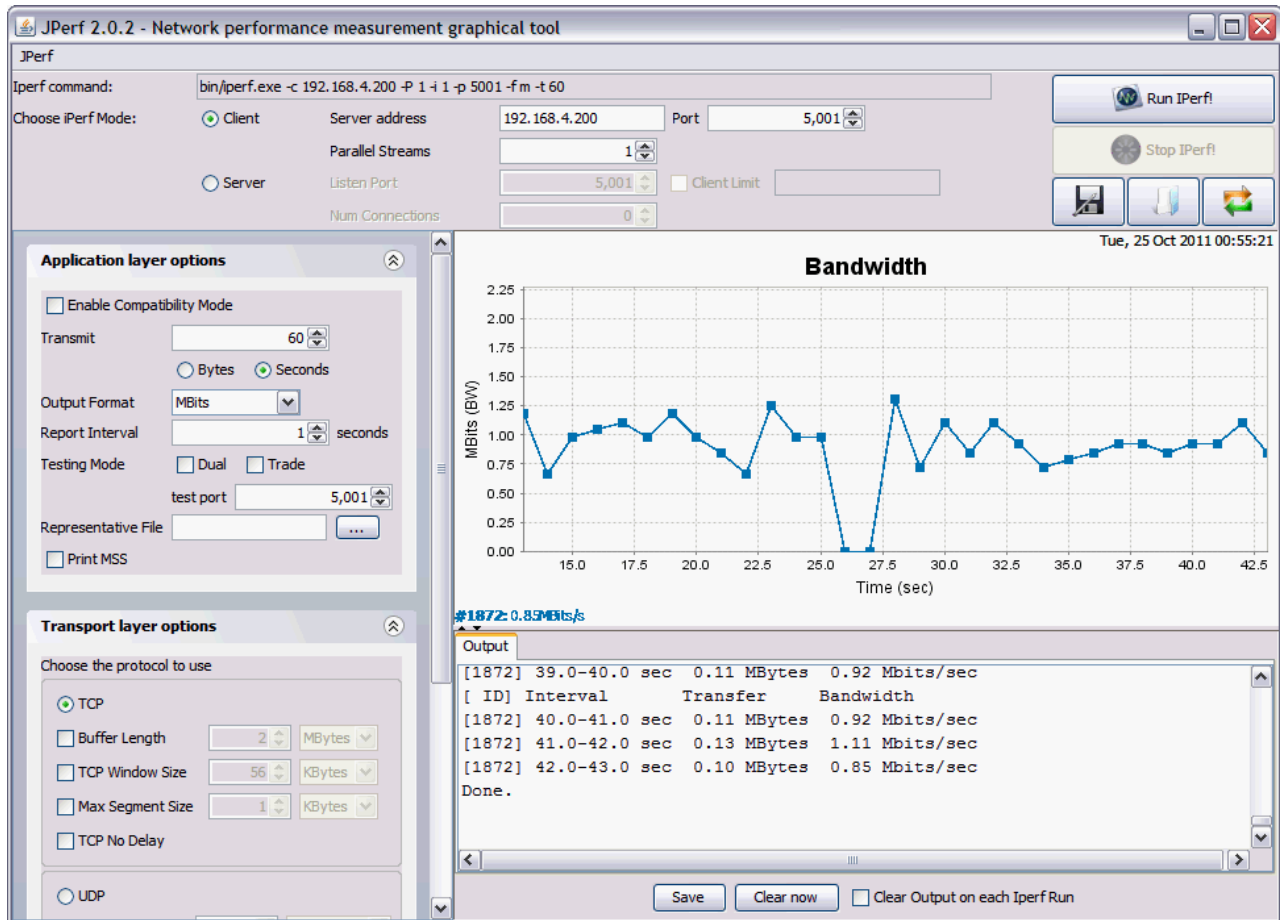


Figure 7-37 JPerf test during live migration of File server

Related publications

The publications listed in this section are considered particularly suitable for a more detailed addressing of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Using IBM Virtualization to Manage Cost and Efficiency*, REDP-4527
- ▶ *IBM Systems Virtualization: Servers, Storage, and Software*, REDP-4396
- ▶ *IBM PowerVM Virtualization Introduction and Configuration*, SG24-7940
- ▶ *System x Virtualization Strategies*, REDP-4480
- ▶ *IBM Systems Director VMControl Implementation Guide on IBM Power Systems*, SG24-7829
- ▶ *IBM Data Center Networking: Planning for Virtualization and Cloud Computing*, SG24-7928
- ▶ *Implementing Microsoft Hyper-V on IBM System x and IBM BladeCenter*, REDP-4481
- ▶ *Integrated Virtual Ethernet Adapter Technical Overview and Introduction*, REDP-4340
- ▶ *IBM BladeCenter Virtual Fabric Solutions*, SG24-7966-00
- ▶ *IBM j-type Data Center Networking Introduction*, SG24-7820
- ▶ *IBM Systems Director VMControl Implementation Guide on IBM Power Systems*, SG24-7829
- ▶ *IBM PowerVM Live Partition Mobility*, SG24-7460

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ Effect of SR-IOV Support in Red Hat KVM on Network Performance in Virtualized Environments
ftp://public.dhe.ibm.com/eserver/benchmarks/wp_SR-IOV_Red_Hat_111810.pdf
- ▶ IEEE standards documents:
 - Architecture overview
<http://www.ieee802.org/1/files/public/docs2010/bg-bottorff-arch-0310-v9.pdf>

- Edge Virtual Bridging discovery protocol
<http://www.ieee802.org/1/files/public/docs2010/bg-krause-100309-v2.pdf>
- Multichannel protocol
<http://www.ieee802.org/1/files/public/docs2010/bg-bottorff-multichannel-0310-v6.pdf>
- Edge TLV Transport Protocol (used to carry VDP Data Units) overview
<http://www.ieee802.org/1/files/public/docs2010/bg-recio-ettp-0310-v0.pdf>
- Virtual Station Interface (VSI) Discovery and Configuration (VDP) overview
<http://www.ieee802.org/1/files/public/docs2010/bg-sharma-evb-VSI-discovery-0310v00.pdf>
- Version 0 proposal document
<http://www.ieee802.org/1/files/public/docs2010/bg-joint-evb-0310-v0.pdf>
- ▶ Wikipedia: Open Virtualization Format
http://en.wikipedia.org/wiki/Open_Virtualization_Format
- ▶ Open Virtualization Format White Paper
http://www.dmtf.org/sites/default/files/standards/documents/DSP2017_1.0.0.pdf
- ▶ Systems Director for physical and virtual environment management
<http://www.ibm.com/systems/virtualization/systemsdirector>
- ▶ Dynamic Infrastructure: Virtualization
<http://www.ibm.com/itsolutions/virtualization>
- ▶ IBM Virtualization: Optimize Your IT Infrastructure
<http://www.ibm.com/systems/virtualization>
- ▶ VMready Technical White Paper
<ftp://public.dhe.ibm.com/common/ssi/ecm/en/qcw03005usen/QCW03005USEN.PDF>
- ▶ VMready Data Sheet
<ftp://public.dhe.ibm.com/common/ssi/ecm/en/qcd03001usen/QCD03001USEN.PDF>
- ▶ VMready Solution Brief
<ftp://public.dhe.ibm.com/common/ssi/ecm/en/qcs03001usen/QCS03001USEN.PDF>
- ▶ VMready: Virtualization-Aware Networking Within and Across Data Centers
<http://www-03.ibm.com/systems/networking/software/vmready.html>
- ▶ P. Congdon, “Virtual Ethernet Port Aggregator Standards Body Discussion”
<http://www.ieee802.org/1/files/public/docs2008/new-congdon-vepa-1108-v01.pdf>
- ▶ R Recio, O Cardona, “Automated Ethernet Virtual Bridging”
http://www.itc21.net/fileadmin/ITC21_files/DC-CAVES/DC-CAVES_-_AutomatedEthernetVirtualBridging.pdf
- ▶ Standardizing Data Center Server-Network Edge Virtualization
<http://public.dhe.ibm.com/common/ssi/ecm/en/qcl12363usen/QCL12363USEN.PDF>
- ▶ Linux information > Blueprints for Linux on IBM systems > Virtualization blueprints: Configuring the network for virtual machines in a data center
<http://publib.boulder.ibm.com/infocenter/lxinfo/v3r0m0/index.jsp?topic=%2F1aa%2Fvswitch%2F1aaivmswitchstart.htm>

- ▶ BLADE Harmony Manager 5.2 Announcement (211-170) - includes Edge Virtual Bridge (802.1Qbg) port profile (VSI) management
http://www-01.ibm.com/common/ssi/rep_ca/0/897/ENUS211-170/ENUS211-170.PDF
- ▶ Edge Virtual Bridge Proposal standards document
<http://www.ieee802.org/1/files/public/docs2010/bg-joint-evb-0410v1.pdf>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Implementing a VM-Aware Network Using VMready

(0.5" spine)
0.475" x 0.873"
250 <-> 459 pages



Implementing a VM-Aware Network Using VMready



Ensuring IT service performance and security

Using virtualization to enhance IT service agility

Improving management to reduce IT support costs

Virtualization allows administrators to create virtual versions of resources such as a hardware platform, operating system, storage device, or network resource. This process allows the creation of multiple versions of a resource on a single physical machine. This configuration allows the advantages of multiple resources while simplifying management tasks and improving use of physical resources. In this way, virtualization can be used to enhance IT service performance, scalability, efficiency, availability, and security. Configuring these virtual machines requires the use of hypervisor software.

However, network switches are not aware of virtual machines. If you run each virtual machine on a dedicated set of servers, performance and security needs can be met by configuring the network settings for each servers. However, this nullifies the main benefits of virtualization. You can get better IT service performance, scalability, efficiency, and availability by creating multiple virtual resources on the same server. For this configuration, you need a way to apply unique network settings for each virtual resource.

IBM VMready is a software solution that supports open standards virtualization based on IEEE 802.1Qbg Edge Virtual Bridging. It allows administrators to create groups of virtual machines, administer them from a central location, and migrate them. VMready works with all major hypervisor software, including VMware, Microsoft Hyper-V, Linux Kernel-based Virtual Machine (KVM), Citrix XenServer, or IBM PowerVM®. It requires no proprietary tagging or changes to the hypervisor software. This IBM Redpaper Redbooks publication helps IT systems and networking professionals to understand IBM VMready technology options. It includes instructions on how to install, tailor, and configure VMready networking solutions.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**

SG24-7985-00

ISBN 0738436852